"GRAMÁTICAS E LINGUAGENS INDICIAIS PROBABILÍSTICAS"

ORION DE OLIVEIRA SILVA

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA (M.Sc.).

Aprovada por:

Presidente

RIO DE JANEIRO
ESTADO DA GUANABARA - BRASIL
AGOSTO DE 1973

À minha mãe

AGRADECIMENTOS

Ao Professor Celso de Renna e Souza, pela sugestão, orientação e constante assistência a este trabalho.

Ao Professor Flávio Roberto Dias Velasco, pelas sugestões dadas em alguns capítulos.

Ao Engenheiro Eugênio Rubens Cardoso Braz, pelas sugestões dadas nos gráficos.

RESUMO

Um tipo de gramática chamado "Gramáticas Indiciais" foi definido por Alfred V. Aho em Austin, Texas, no ano de 1967. As linguagens geradas por gramáticas indiciais são chamadas linguagens indiciais. As classes de linguagens indiciais incluem, propriamente , as classes de linguagens Livre de Contexto e é um subconjunto próprio de classe de linguagens sensíveis ao contexto.

O nosso objetivo neste trabalho é atribuir probabilidades às produções das gramáticas indiciais e estudar a consistência das linguagens geradas por essas gramáticas.

ABSTRACT

A new type of grammar for generating formal languages, called indexed grammar, was presented by Alfred V. Aho, in Austin , Texas, 1967. The languages generated by indexed grammars are called indexed languages. The class of languages generated by indexed grammars properly includes all context free languages and is a proper subset of the class of context sensitive languages.

The objective in this work is to assign probability to each production of the indexed grammar and study the consistency of the indexed language generated by it.

ÍNDICE

CAPÍTULO I

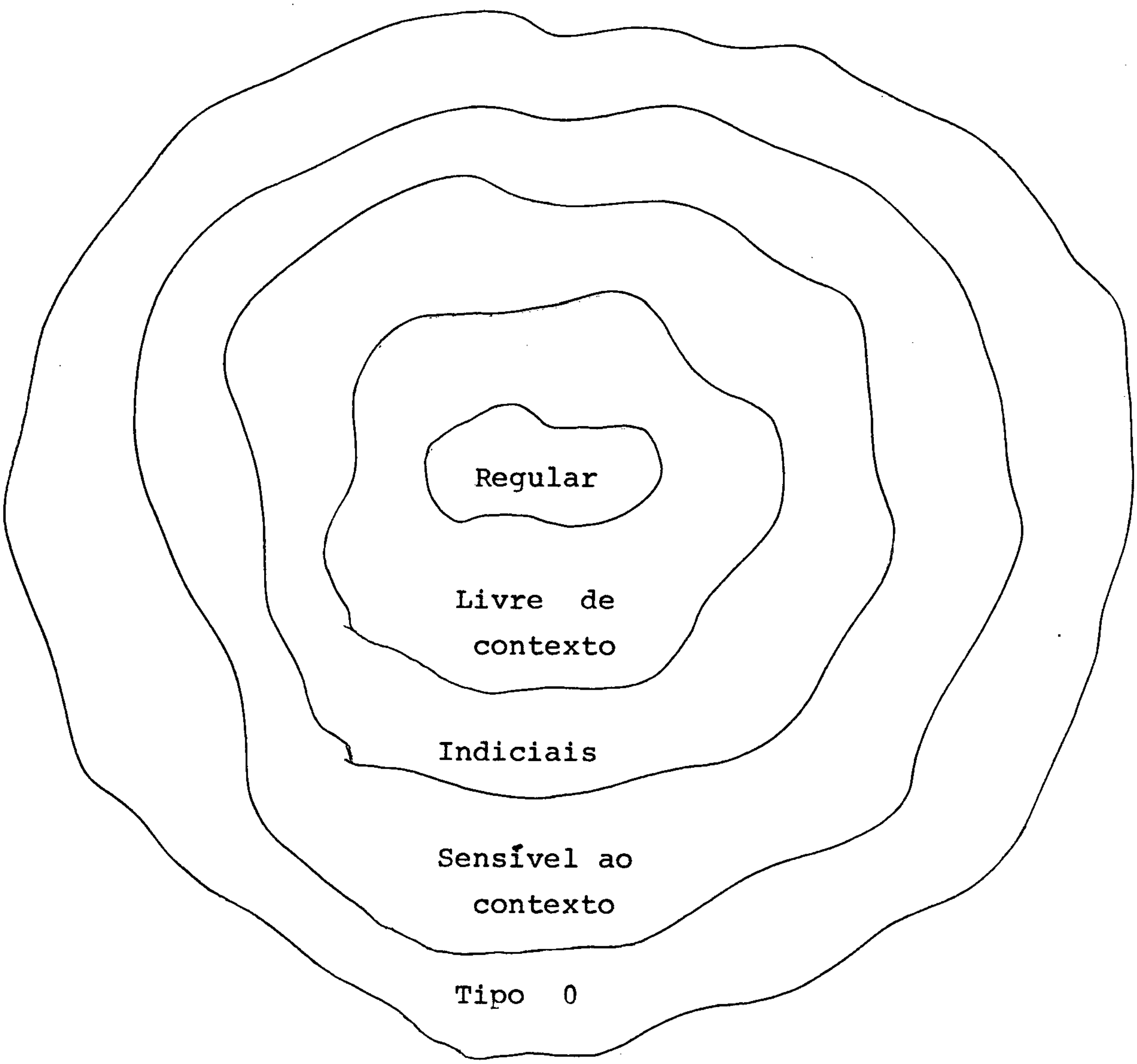
I N T R O D U Ç Ã O

O conceito de máquinas e processos tem sido estudado em Processo de Markov [1] , processo linearmente independente [1] , e muitas variedades de autômato probabilístico [13] , [14] , [15] , [16].

A noção de função sequencial discreta foi introduzida por Chomsky [17] . Como as saídas de um processo de estado de Markov é um processo linearmente dependente, assim as saídas de uma gramática probabilística é uma linguagem probabilística. Há um relacionamento estimado entre classes de linguagens formais e classes de autômato [1] , [5] , [9] e muitas variedades de autômato probabilístico tem sido propostas. Há muitos conceitos de linguagem probabilística na literatura; o conceito usado neste trabalho é o usado por Booth [2] e Ellis [3].

Recentemente, vários trabalhos tem definido um grande número de interessantes classes de linguagens, definidas recursivamente, maiores do que as classes de linguagens livres de contexto . Um exemplo disto são as gramáticas programadas [6]. Parte dessas classes maiores de linguagens origina-se da insuficiência das gramáticas livres de contexto para especificar toda a estrutura sintática encontrada em muitos algorítmos de linguagens de programação dos nossos dias, tal como ALGOL, por exemplo [18].

1.1 - Gráfico das Linguagens



CAPÍTULO II

GRAMÁTICAS INDICIAIS E GERAÇÃO DE SENTENÇAS

Neste capítulo é dada a definição de gramáticas indiciais, linguagens indiciais e vários exemplos de gramáticas indiciais, como também o conceito de "flag" [7].

2.1 - Gramáticas Indiciais

Definição 2.1 - Uma gramática indicial é uma quíntupla $G=\langle N,T,F,P,S\rangle$ tal que:

(a) N é um conjunto finito não vazio de símbolos, chamado alfabeto não-terminal.

(b) T é um conjunto finito não vazio de símbolos, chamado alfabeto terminal.

(c) F é um conjunto de elementos da forma $\left[(A_1,X_1),(A_2,X_2),\ldots,(A_K,X_K) \right]$, onde $A_i \varepsilon N$ e $X_i \varepsilon (N \cup T)^*$ para todo $1 \leq i \leq K$. Um elemento f de F é chamado índice ou flag. Um par ordenado (A_i,X_i) de f será usualmente escrito com $A_i \to X_i$ e será chamado de produção indicial.

(d) P é um conjunto finito de pares ordenados da forma (A,α) com $A \varepsilon N$ e $\alpha \varepsilon (NF^* \cup T)^*$. Tal par será usualmente escrito $A \to \alpha$ e será chamado de produção.

(e) S é o símbolo inicial $S \varepsilon N$.

<u>Nota</u>: Seja T um alfabeto. T^+ é o conjunto de todas as listas de tamanho finito de símbolos de T, excluindo o Λ (lista vazia). $T = T^+ \cup \{\Lambda\}$.

Seja $G = \langle N,T,F,P,S \rangle$ uma gramática indicial; grosseiramente falando, uma derivação de G é uma sequência de listas α_1, $\alpha_2,\ldots,\alpha_n$ com $\alpha_i \varepsilon (NF^* \cup T)^*$ no qual α_{i+1} é derivado de α_i por aplicação de uma produção ou produção indicial. Exceto da maneira pela qual índices e listas de índices são manipulados, uma derivação de G se processa exatamente da mesma maneira como numa gramática Livre de Contexto [5]. Entretanto em uma derivação de G, cada não-terminal de N pode ser imediatamente seguido por uma lista de índices de F^*. Uma lista de símbolos da forma $A\mathit{I}$, onde $A \varepsilon N$ e $\mathit{I} \varepsilon F^*$ será chamado não-terminal indiciado. Uma lista α onde $\alpha \varepsilon (NF^* \cup T)^*$ será chamado forma sentencial.

Por exemplo, se $A\,f\,\mathit{I}$ é um não-terminal indiciado, o índice f contém a produção indicial $A \rightarrow aBC$, então $A\,f\,\mathit{I}$ pode derivar a forma setencial $aB\mathit{I}C\mathit{I}$ em uma derivação de G. Note que em cada passo o índice f é removido na próxima linha da derivação e a lista indicial I se distribui sobre os não-terminais B e C mas não sobre o símbolo terminal a.

Se $A\mathit{I}$ é um não-terminal indiciado e $A \rightarrow aB\eta C\theta b$ é uma produção de P, então de $A\mathit{I}$ se pode derivar a forma sentencial $aB\eta\mathit{I}C\mathit{I}b$ em uma derivação de G. Aqui a lista indicial I se distribui sobre o não-terminal indiciado $B\eta$ e $C\theta$ mas não sobre os símbolos terminais a e b na produção.

2.2 - Forma Sentencial Gerada (derivada) Diretamente

Uma forma sentencial β é dita gerada diretamente (ou derivada diretamente) de uma forma sentencial α , escrevemos $\alpha \underset{G}{\rightarrow} \beta$ se e somente se:

1) $\alpha = \gamma A I \delta$ com $\gamma, \delta \in (NF^* \cup T)^*$, $A \in N$ e $I \in F^*$

$A \rightarrow X_1 \eta_1 X_2 \eta_2 \ldots X_K \eta_K$ com $\eta_i \in F^*$ e $i=1,\ldots,K$, $X_i \in N \cup T$.

$\beta = \gamma X_1 \theta_1 X_2 \theta_2 \ldots X_K \theta_K \delta$

$$\theta_i = \begin{cases} \eta_i & \text{se } X_i \in N, \quad i=1,\ldots,K \\ \Lambda \ \varepsilon \ X_i \in T & \Lambda \text{ (lista vazia)} \end{cases}$$

2) $\alpha = \gamma A f I \delta$ com $\gamma, \delta \in (NF^* \cup T)^*$, $A \in N$, $I \in F^*$ e $f \in F$

$A \rightarrow X_1 X_2 \ldots X_K$ produção indicial com índice em f .

$\beta = \gamma X_1 \theta_1 X_2 \theta_2 \ldots X_K \theta_K \delta$

$$\theta_i = \begin{cases} I & \text{se } X_i \in N \\ \Lambda & \text{se } X_i \in T \end{cases}$$

No caso 1) e 2) o não-terminal A é chamado expandido. No caso 2) dizemos que o índice f é consumido pelo não - terminal A.

<u>Observação</u>: Um símbolo terminal nunca tem uma lista de índices imediatamente seguido em nenhum ponto da derivação.

Se em cada forma sentencial α_i , o não-terminal mais à esquerda é expandido para gerar diretamente a lista α_{i+1} para

$0 \leq i \leq n$, então a sequência de listas 1) é chamada de derivação mais à esquerda. Mostra-se facilmente que se existe uma derivação β de α , então existe uma derivação mais à esquerda de β de α [7] . A palavra derivação de agora em diante terá o significado de derivação mais à esquerda, a não ser que se especifique o contrário.

Exemplo 2.1 - Seja $G_1 = \langle \{S,A,B\}, \{a,b,c\}, \{f,g\}, P,S \rangle$ onde P consiste das produções:

$S \rightarrow aAfc$

$A \rightarrow aAgc$

$A \rightarrow B$

e onde

$f = [B \rightarrow b]$

$g = [B \rightarrow bB]$

Aplicando a primeira produção uma vez, a segunda produção $n-1$ vezes, $n \geq 1$, e a terceira produção uma vez, temos:

$S \rightarrow aAfc \rightarrow aaAgfcc \rightarrow \ldots \rightarrow a^nAg^{n-1}fc^n \rightarrow a^nBg^{n-1}fc^n$. Então, expandindo B por índices consumidos, temos:

$Bg^{n-1}f \rightarrow bBg^{n-2}f \rightarrow \ldots \rightarrow b^{n-1}Bf \rightarrow b^n$

Então

$S \xrightarrow{2n+1} a^nb^nc^n$

Então

$L(G_1) = \{a^nb^nc^n \ / \ n \geq 1\}$

Exemplo 2.2 - Seja $G_2 = \langle \{S,T,A,B,C\}, \{a,b\}, \{f,g\},P,S \rangle$

onde P contém as produções

$S \rightarrow Tf$

$T \rightarrow Tg$

$T \rightarrow ABC$

e onde

$f = [A \rightarrow a,\ B \rightarrow b,\ C \rightarrow b]$

$g = [A \rightarrow aA,\ B \rightarrow bBCC,\ C \rightarrow bC]$

O que é fácil de mostrar que:

$L(G_2) = \{a^n b^{n^2} a^n \ / \ n \geq 1\}$

Exemplo 2.3 - Seja $G_3 = \langle \{S,B,T\}, \{a,b,c\}, \{f,g\}, P,S \rangle$

onde P contém as produções:

$S \rightarrow aTfb$

$T \rightarrow aTg$

$T \rightarrow B$

e onde

$f = [B \rightarrow c]$

$g = [B \rightarrow Bb]$

É fácil mostrar que:

$L(G_3) = \{a^n c b^n \ / \ n \geq 1\}$

Exemplo 2.4 - Seja $G_4 = \langle \{S,B,T\}, \{0,1\}, \{f,g\}, P,S \rangle$

onde P contém as produções:

$S \rightarrow oTf$

$T \rightarrow oTg$

$T \to B$

e onde

$f = [B \to 1]$

$g = [B \to oB]$

É fácil mostrar que:

$L(G_4) = \{o^n 1 \ / \ n \geq 1\}$ é regular.

Exemplo 2.5 - Seja $G_5 = \langle \{S,V,A,B,C,B_1,B_2,B_3\}, \{a,b,c,\ldots \wedge, \vee, (\ ,\ ), \sim\}, \{f,g,h\}, P, S \rangle$

onde P contém as produções:

$S \to V$

$V \to \sim Vf$

$V \to B_1$

$V \to (VBVAg)$

$V \to B_2$

$V \to (VCVAh$

$V \to B_3$

e onde

$f = [B_1 \to a/b/c/\ldots]$

$g = [B_2 \to a/b/c/\ldots,\ B \to \vee\ ,\ A \to\ \ )\ ]$

$h = [B_3 \to a/b/c\ \ldots,\ C \to \wedge\ ,\ A \to\ \ )\ ]$

Podemos mostrar que esta gramática indicial gera o conjunto das proposições.

CAPÍTULO III

ÁRVORE DE DERIVAÇÃO EM GRAMÁTICAS INDICIAIS

3.1 - Árvore de Derivação

Uma árvore de derivação de uma gramática indicial $G = \langle N,T,F,P,S \rangle$ será uma árvore com raiz e com nós rotulados. Cada nó consiste de uma n -pla de inteiros positivos da forma $(i_1,i_2,\ldots,i_r)$ onde $r \geq 1$, e os ramos da árvore serão pares ordenados de nós da forma $((i_1,\ldots,i_r),(i_1,\ldots,i_r,i_{r+1}))$. Cada nó da árvore de derivação será rotulado por um símbolo não-terminal indiciado ou símbolo terminal de $NF^* \cup T$.

Exemplo 3.1 - $AI \equiv$ não-terminal indiciado, $I \in F$.

Seja ℓ o número máximo de não-terminais e terminais do lado direito de alguma produção de P. A cada nó $n=(i_1,\ldots,i_r)$, podemos associar a ℓ-ésima fração $0.i_1i_2 \ldots i_r$ que chamamos número do nó para n. Nós dizemos que $n_1 < n_2$(ou $n_1 \leq n_2$) se o número de nós para n_1 é menor do que (ou menor ou igual) para n_2.

Suponhamos D uma árvore de derivação e $\{\alpha_1ni_1, \alpha_2ni_2, \ldots, \alpha_Kni_K\}$, o conjunto de folhas rotuladas de D, arranjadas de tal maneira que $ni_j < ni_{j+1}$ (uma folha é um nó $(i_1,\ldots,i_s)$ de D tal que não há nó de D da forma $(i_1,\ldots,i_s,j)$ para alguns j) para $i \leq j < K$. Cada nó é rotulado por α_i com $\alpha_i \in NF^* \cup T$. A lista $\alpha_1,\alpha_2,\ldots,\alpha_K$ é chamada campo de D.

Uma árvore de derivação de uma gramática indicial $G = \langle N,T,F,P,S\rangle$ com raiz $AI_o, I_o \in F^*$, é um conjunto de nós $A \in N$ rotulados formalmente definidos como segue:

1. O conjunto contendo exatamente o nó rotulado $AI_o(1)$ é uma árvore de derivação.

2. Suponhamos uma árvore de derivação com campo $\beta BI\gamma$ onde $\beta,\gamma \in (NF^* \cup T)^*$ é um nó rotulado de folha $(i_1,i_2,\ldots,i_r)$. Se $B \to X_1\eta_1 X_2\eta_2 \ldots X_K\eta_K$ pertence a P, então $D' = D \cup \{X_1\theta_1 m_1, X_2\theta_2 m_2, \ldots, X_K\theta_K m_K\}$ é uma árvore de derivação onde $1 \leq j \leq K$

$$\theta_j = \begin{cases} \eta_j I & \text{se } X_j \in N \\ \Lambda & \text{, caso contrário e } m_j = (i_1,i_2,\ldots,i_r,j) . \end{cases}$$

Note que o campo de D' é:

$$\beta X_1\theta_1 X_2\theta_2,\ldots,X_K\theta_K\gamma I .$$

3. D é uma árvore de derivação de G com raiz AI_o se e somente se ela segue 1 e 2 .

Isto mostra claramente que, dada uma gramática indicial $G = \langle N,T,F,P,S\rangle$ para cada derivação de G existe uma árvore de derivação correspondente e para cada árvore de derivação de G, existe naturalmente uma derivação (e exatamente uma derivação mais à esquerda).

Teorema 3.1 - Dada uma gramática indicial $G = \langle N,T,F,P,S\rangle$, $AI \overset{*}{\underset{G}{\Rightarrow}} \alpha$ se e somente se existe uma árvore de derivação de G com raiz AI e campo α . [7]

Nós examinaremos nessa seção um exemplo de uma árvore de derivação de uma gramática indicial. Por conveniência de notação, cada nodo da árvore de derivação será representado somente pelos seus nodos rotulados.

Exemplo 3.2 - Seja $G_2 = \langle \{S,T,A,B,C\},\{a,b\},\{f,g\},P,S \rangle$

onde P contém:

$S \rightarrow Tf$

$T \rightarrow Tg$

$T \rightarrow ABA$

e onde

$f = [A \rightarrow a,\ B \rightarrow b,\ C \rightarrow b]$

$g = [A \rightarrow aA,\ B \rightarrow bBCC,\ C \rightarrow bC]$

É fácil mostrar que $L(G_2) = \{a^n b^{n^2} a^n \ / \ n \geq 1\}$. $a^2b^4a^2$ tem a árvore de derivação mostrada na figura 3.1.

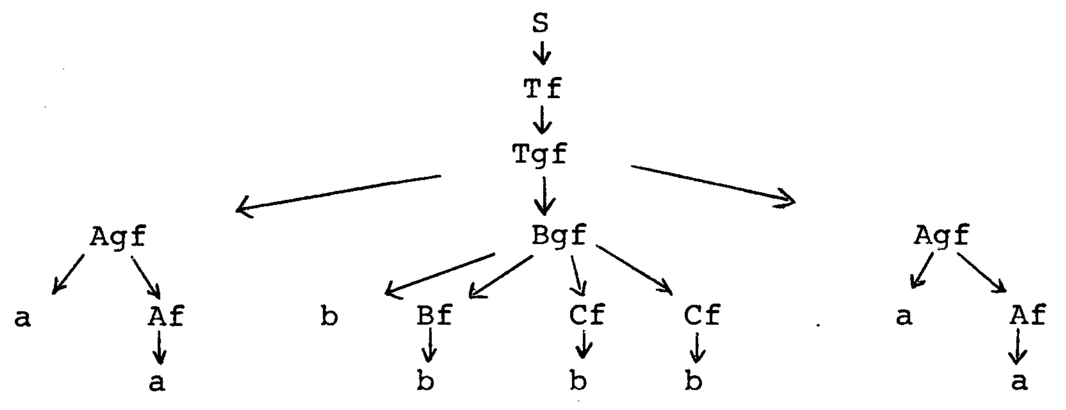


Figura 3.1

CAPÍTULO IV

PROPRIEDADE DE FECHAMENTO

Este capítulo tem como finalidade mostrar o fechamento das linguagens indiciais na forma reduzida, equivalência de linguagens indiciais e a construção de um novo tipo de Autômato denominado Transdutor Finito Não-Determinístico [7].

4.1 - Transdutor Finito Não-Determinístico

Definição 4.1 - Transdutor finito não determinístico TFN. É a sêxtupla $M = \langle Q,T,\Sigma,\delta,g_o,F \rangle$, onde:

a) Q - conjunto finito de estados

b) T - símbolos de entrada

c) Σ - símbolos de saída

d) δ - mapeador de $Qx(T \cup \{\Lambda\})$ em um subconjunto finito de $Qx\Sigma^*$. Se δ é de QxT em $Qx\Sigma^*$, então M é um transdutor finito determinístico

e) $g_o \varepsilon Q$ - símbolo inicial

f) $F \subseteq Q$ - conjunto de estados finais

Se $\hat{\delta}$ é um mapeador expandido de QxT^* em $Qx\Sigma^*$ definido como:

i) $\hat{\delta}(q,\Lambda)$ contém (q,Λ)

ii) Se $\hat{\delta}(q_1,w)$ contém (q_2,x) e $\delta(q_2,a)$ contém (q_3,y), então $\hat{\delta}(q_1,w_a)$ contém (q_3,x_y) para $w \varepsilon T^*$, $a \varepsilon T \cup \{\Lambda\}$ e $x,y \varepsilon \Sigma^*$.

Um transdutor finito não-determinístico $M = \langle Q, T, \Sigma, \delta, g_o, F \rangle$ pode induzir um mapeamento em uma linguagem $L \subseteq T^*$ da seguinte maneira:

Para $W \varepsilon T^*$ $M(W) = \{x/\hat{\delta}(g_o, w)$ contém (p,x) para algum $p \varepsilon F\}$

$$M(L) = \bigcup_{W \varepsilon L} M(W)$$

Podemos definir o mapeador inverso

$$M^{-1}(L') = \{W/M(W) \varepsilon L'\}$$

Dado um TFN $M = \langle Q, T, \Sigma, \delta, g_o, F \rangle$ um TFN M' pode ser construído a partir de M tal que $M'(L) = M^{-1}(L)$ para todo $L \subseteq \Sigma^*$.

4.2 - Gramática Indicial na Forma Reduzida

Definição 4.2 - Uma gramática indicial $G = \langle N, T, F, P, S \rangle$ é dita na forma reduzida se:

a) Cada produção indicial em cada índice F é da forma $A \rightarrow B$ onde $A, B \varepsilon N$.

b) Cada produção P é de uma das formas:

1) $A \rightarrow BC$ $\quad A, B, C \varepsilon N$

2) $A \rightarrow Bf$ com $f \varepsilon F$

3) $A \rightarrow a$ $\quad a \varepsilon T \cup \{\Lambda\}$ [7]

4.3 - Equivalência de Gramáticas Indiciais

Definição 4.3 - Duas gramáticas indiciais G_1 e G_2 são ditas equivalentes se $L(G_1) = L(G_2)$ [7].

Teorema 4.1 - Dada uma gramática indicial $G = \langle N,T,F,P,S \rangle$, uma gramática indicial $G' = \langle N',T,P',S,F' \rangle$ equivalente na forma reduzida pode ser construída a partir de G. A demonstração deste teorema acima está em [7] .

4.4 - Fechamento de Linguagens Indiciais

Lema 4.1 - A classe de linguagens indiciais é fechada sobre o mapeador TFN [7] .

Demonstração: Seja $G = \langle N,T,F,P,S \rangle$ uma gramática indicial na forma reduzida e seja $M = \langle Q,T,\Sigma,\delta,q_0,\Sigma \rangle$ um TFN. Construiremos uma gramática indicial $G' = \langle N',\Sigma,F',P',S' \rangle$, tal que $L(G') = M(L(G))$.

O não-terminal de M' será da forma (p,X,q) onde $p,q \in Q$ e $X \in N \cup T \cup \{\Lambda\}$. O conjunto de produções P' é construído como segue:

1 - Se $A \to BC$ é de P, então P' contém o conjunto de produções $(p,A,q) \to (p,B,r)(r,C,q) \; \forall \; p,q,r \in Q$.

2 - Se $A \to Bf$ é de P, então P' contém o conjunto de produções $(p,A,q) \to (p,B,q)f' \; \forall \; p,q \in Q$.

O índice f' contém a produção indicial $(r,C,s) \to (r,D,s) \; \forall \; r,s \in Q$ se e somente se f contém a produção indicial $C \to D$.

3 - Se $A \to a$ é de P, então P' contém o conjunto de produções $(p,A,q) \to (p,a,q)$ para todo $p,q \in Q$.

4 - P' também contém as produções

$(p,a,q) \to (p,a,r)(r,\Lambda,q)$

$(p,a,q) \to (p,\Lambda,r)(r,a,q)$

$(p,\Lambda,q) \to (p,\Lambda,r)(r,\Lambda,q)$

para todo a de T e $p,q,r \in Q$.

5 - P' contém a produção terminal $(p,a,q) \to x$ e $\delta(p,a)$ contém (q,x) para $a \in T \cup \{\Lambda\}, x \in \Sigma^*$.

6 - Finalmente, P' contém as produções iniciais

$S' \to (q_o,S,p) \quad p \in K$.

Nós agora mostraremos que $(p,A,q)f_1' \dots f_j' \xrightarrow[G']{*} x$,$j \geq 0$, $x \in \Sigma^*$, se e somente se $Af_1 \dots f_j \xrightarrow[G]{*} w$ e $\hat{\delta}(p,w)$ contém (q,x) para algum $w \in T^*$.

A demonstração será por indução na extensão de uma derivação.

$\Longrightarrow$ (*) Se $(p,A,q)f_1' \dots f_j' \xrightarrow[G"]{K} x$ é alguma derivação de tamanho K, então $Af_1 \dots f_j \xrightarrow[G]{*} w$ e $\hat{\delta}(p,w)$ contém (q,x) por algum w de T^*. Suponhamos que (*) seja verdade para toda derivação de tamanho $K<m$. Isto é verdade para derivação de tamanho 1. Consideremos agora uma derivação $(p,A,q)f_1' \dots f_j' \xrightarrow[G']{m} x$ de tamanho $m,m>1$.Estas derivações podem ser de quatro diferentes formas.

(i) $(p,A,q)f_1' \dots f_j' \xrightarrow[G']{} (p,B,r)f_1' \dots f_j'(r,C,q)f_1' \dots f_j'$

$\xrightarrow[G']{m_1} x_1(r,C,q)f_1' \dots f_j'$ onde $m_1 < m$

$\xrightarrow[G']{m_2} x_1x_2$ onde $m_2 < m$

Da hipótese indutiva, nós temos a seguinte derivação em G:

$$Af_1 \ldots f_j \xrightarrow[G]{} Bf_1 \ldots f_jCf_1 \ldots f_j \xrightarrow[G]{*} w_1Cf_1 \ldots f_j \xrightarrow[G]{*} w_1w_2$$

Também $\hat{\delta}(q,w_1)$ contém (r,x_1) e $\hat{\delta}(r,w_2)$ contém (q,x_2) da hipótese indutiva. Então $\hat{\delta}(q,w)$ côntém (q,x) onde $w = w_1w_2$ e $x = x_1x_2$.

(ii) $(p,A,q)f_1' \ldots f_j' \xrightarrow[G']{} (p,B,q)f'f_1' \ldots f_j' \xrightarrow[G']{m'} x$

com $m' < m$.

Em G nós temos a seguinte derivação

$Af_1 \ldots f_j \xrightarrow[G]{} Bff_1 \ldots f_j \xrightarrow[G]{*} w$ e da hipótese indutiva

$\hat{\delta}(p,w)$ contém (q,x) .

(iii) $(p,A,q)f_1' \ldots f_j' \xrightarrow[G']{} (p,a,q)f_1' \ldots f_j' \xrightarrow[G']{m'} x$

com $m' < m$.

É óbvio mostrar que $(p,a,q) \xrightarrow[G']{*} x$ com $a \in T \quad \{\Lambda\}$ se e somente se $\hat{\delta}(p,a)$ contém (q,x). Consequentemente, em G temos: $Af_1 \ldots f_j \xrightarrow[G]{} a$ e $\hat{\delta}(p,a)$ contém (q,x) .

(iv) $(p,A,q)f_1'f_2' \ldots f_j' \xrightarrow[G']{} (p,B,q)f_2' \ldots f_j' \xrightarrow[G']{m'} x$

com $m' < m$.

Aqui o índice f_1' contém a produção indicial

$(p,A,q) \longrightarrow (p,B,q)$ e é consumido na primeira linha da deriva - ção. Em G temos a derivação

$$Af_1f_2 \ldots f_j \xrightarrow[G]{} Bf_2 \ldots f_j \xrightarrow[G]{*} w \quad \text{para algum } w \in T^* \quad \text{e}$$

da hipótese indutiva $\hat{\delta}(p,w)$ contém (q,w) .

Esses quatro casos representam todos os possíveis caminhos nos quais a primeira expansão na derivação em G' pode pros̲seguir.

$\Longleftarrow$ Na recíproca, uma demonstração semelhante por indução no tamanho da derivação em G.

Se $Af_1 \ldots f_j \xrightarrow[G]{*} w$ e $\hat{\delta}(p,w)$ contém (q,x) então $(p,A,q)f_1' \ldots f_j' \xrightarrow[G']{*} x$.

Então temos: $s' \xrightarrow[G']{} (q_o,s,p) \xrightarrow[G']{} x$ para p de K ,se e somente se $s \xrightarrow[G]{*} w$, $w \in T^*$ e $\hat{\delta}(q_o,w)$ contém (p,x). Então $L(G') = M(L(G))$.

Desde que TFN inverso mapeia também o mapeador TFN, temos que: "A classe de linguagens indiciais é fechada sobre o mapeador TFN inverso" [7].

CAPÍTULO V

PROPRIEDADES DE LINGUAGENS E GRAMÁTICAS PROBABILÍSTICAS

Um considerável conhecimento já existe ([9], [5], [18], [22], [24], [1] e [25]) sobre propriedades sintáticas de linguagens formais. Alguns trabalhos foram realizados ([2], [10], [11], [12], [13], [14], [15], [16], [19], [20], [21], [26] e [27]) no sentido de desenvolver o relacionamento formal que pode ser usado para descrever as propriedades gramaticais e semânticas de uma linguagem. Uma das razões para isso é a possibilidade da estrutura gramatical de uma linguagem formal ser descrita por um conjunto de regras determinísticas, enquanto que as propriedades semânticas de linguagem podem ser bastante ajudadas por técnicas probabilísticas.

5.1 - Gramáticas Formais e Convenção Usada

Os símbolos N,T,P,S são os mesmos conjuntos que definimos anteriormente (não-terminais, terminais, conjunto de produções e símbolo inicial, respectivamente).

As letras gregas são listas pertencentes a V^* , onde $V = N \cup T$.

Toda produção de P será representada por $I \to \alpha$, onde I é chamada premissa e α é chamada consequência.

Definição 5.1 - Uma gramática G é uma quádrupla $G = \langle N,T,P,S \rangle$ onde N,T,P e S são os conjuntos definidos anteriormente.

Uma gramática é definida como sensível ao contexto, Livre do Contexto ou regular , dependendo da forma da produção sobre o conjunto P.

Definição 5.2 - Gramática sensível ao contexto é uma gramática em que as produções de P são da forma:

$$I \rightarrow \alpha \Leftrightarrow |I| \leq |\alpha|$$

Definição 5.3 - Uma gramática é livre de contexto se todas as produções são da forma $I \rightarrow \alpha$ onde $I \in N$.

Definição 5.4 - Uma gramática é regular quando as produções são da forma:

$$A \rightarrow aB \quad \text{ou} \quad A \rightarrow a \qquad A,B \in N \quad \text{e} \quad a \in T$$

Definição 5.5 - Uma linguagem de uma gramática G, L(G) é definida por:

$$L(G) = \{x/S \xrightarrow{*} x \ , \ x \in T\} \quad \text{onde} \quad S \xrightarrow{*} x$$

significa que existe uma sequência de produções de P que possibilita a geração da sequência x começando com o símbolo inicial S.

5.2 - Probabilidade Associada com Linguagem

Definição 5.6 - Uma linguagem $L \subset T^*$ é uma "linguagem probabilística se e somente se existe uma medida $p(x)$ para cada $x \in L$, tal que:

a) $0 \leq p(x) \leq 1$

b) $\sum_{x \in L} p(x) = 1$

Como definimos, $p(x)$ depende de muitos fatores.

Se L é uma linguagem com um número infinito de listas, então não é possível listar todos os valores de $p(x) \ \forall \ x \in L$. É necessário então desenvolver um algorítmo que possa ser usado para associar um valor $p(x)$ para cada $x \in L$. O primeiro caminho é descrever este algorítmo pelas propriedades sintáticas da linguagem. Um segundo caminho é expressar $p(x)$ como uma função de alguma característica estrutural que é associada com cada $x \in L$.

5.3 - Construção de Probabilidade Associada com Produção

Definição 5.5 - Uma gramática probabilística é definida pela quádrupla $G = \langle T,N,R,S \rangle$ onde T,N e S são já conhecidos e R é o conjunto de produções com probabilidade p_i, $0 \leq p_i \leq 1$.

Para algum $x \in L$ gerado por uma gramática G existe uma ou mais sequências de produções que podem ser usadas para gerar x. Se α_i são as listas intermediárias na r-ésima sequência de geração que pode ser usada para gerar x. Essa sequência é indicada por:

$$\alpha \xrightarrow{r_{p\alpha_1}} \alpha_1 \xrightarrow{r_{p\alpha_2}} \alpha_2 \longrightarrow \ldots \xrightarrow{r_{p\alpha_n}} \alpha_n = x$$

onde $r_{p\alpha_i}$ é a regra de produção usada para ir de α_{i-1} à α_i na r-ésima sequência de geração.

Definição 5.6 - A probabilidade de x é definida por:

$$p(x) = \sum p(r_{p\alpha_1})\, p(r_{p\alpha_2} / r_{p\alpha_1}) \ldots p(r_{p\alpha_n}/r_{p\alpha_1} r_{p\alpha_2} r_{p\alpha_3} \ldots r_{p\alpha_{n-1}})$$

onde a somatória se estende sobre todas as produções, tal que $s \overset{*}{\longrightarrow} x$ e $p(r_{p\alpha_i} / r_{p\alpha_1} \ldots r_{p\alpha_{i-1}})$ é a probabilidade condicional que a produção $r_{p\alpha_i}$ seja a i-ésima produção usada na r-ésima geração distinta de x .

A linguagem L pode ser gerada por um número de gramáticas diferentes e as probabilidades podem ser definidas por uma infinidade de definições. Seja R representando um conjunto de regras que define como as probabilidades são associadas com a produção de uma gramática G que gera a linguagem L.

Definição 5.7 - R é uma representação probabilística consistente para a linguagem L gerada pela gramática $G \Longleftrightarrow \sum p(x) = 1$

Definição 5.8 - R é uma representação probabilística sem restrição da linguagem L(G) se e somente se:

$$p(r_{p\alpha_i} / r_{p\alpha_1} \ldots r_{p\alpha_{i-1}}) = p(r_{p\alpha_i})$$

5.4 - Atribuição de Probabilidade para Gramáticas "Livre de Contexto"

Em uma linguagem livre de contexto, todas as produções são da forma:

$$I \rightarrow \alpha \quad \text{onde} \quad I \in N \quad \text{e} \quad \alpha \in (N \cup T)^* - \Lambda$$

Vamos investigar a forma que R deve tomar para que a representação probabilística para a linguagem livre de contexto seja consistente. Vamos para isto usar a teoria de Galton-Watson para "Processos Múltiplos de Ramificação" [4] .

Para aplicar a teoria de "Processos de Ramificação" é necessário estabelecer a idéia de nível do processo de geração usado para gerar uma lista $x \in L(G)$.

O nível zero de uma geração será tomado como S. O nível 1 será tomado como β_1 onde β_1 é a lista gerada pela produção $S \rightarrow \beta_1$. O segundo nível será correspondente à lista β_2 que é obtida de β_1 por aplicarmos as produções apropriadas para todo elemento não-terminal de β_1. Se β_1 não contém nenhum não-terminal, o processo é terminado. Estendendo essa idéia, o i-ésimo nível a lista β_i é definida como a lista obtida da lista β_{i-1} por aplicarmos produções apropriadas para todo elemento não-terminal de β_{i-1}, portanto não correspondendo à derivação mais à esquerda.

Exemplo 5.1 - Seja $G = \langle S,N,T,P \rangle$ e $N = \{S,A,B\}$, $T = \{0,1\}$

$$P = \{S \rightarrow 0AB,\ A \rightarrow BA,\ A \rightarrow 1BB,\ B \rightarrow 00B,\ B \rightarrow 0\}$$

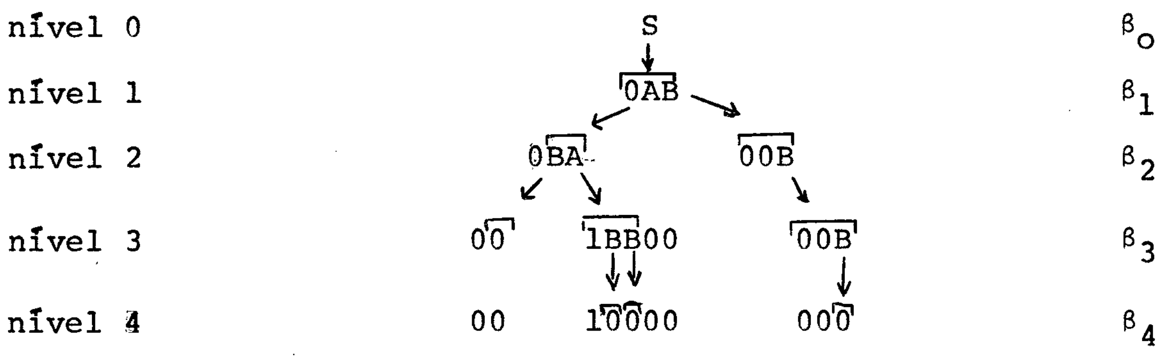


Figura 5.1

Como todo elemento não-terminal é considerado simultaneamente para ir do nível $i-1$ para o nível i, somente as probabilidades associadas com as produções necessitam ser consideradas. Essas probabilidades são arbitrárias.

Seja $P = \Gamma A_1 \quad \Gamma A_2 \quad \ldots \quad \Gamma A_K$ a partição de P na classe de equivalência, tal que duas produções estão na mesma classe se e somente se elas tem a mesma premissa. Para cada ΓA_j definimos probabilidades para as regras de produção $A_j \to \alpha$ e denominamos por $\{p(\alpha/A_j)\}$, onde $\sum\limits_{\Gamma A_j} p(\alpha/A_j) = 1$.

Seja $r_{j,\ell}(\alpha)$ o número de vezes que a variável A_ℓ aparece na lista α da produção $A_j \to \alpha$.

Definição 5.9 - Para cada ΓA_j $j=1,\ldots,K$ definimos os K argumentos gerando funções $f_j(s_1,s_2,\ldots,s_K)$ com $f_j(s_1,s_2,\ldots,s_K) =$

$$\sum_{rA_j} p(\alpha/A_j) s_1^{r_{i,1}(\alpha)} \cdots\cdots s_K^{r_{i,K}(\alpha)}$$

Exemplo 5.2 - Seja $G = \langle S,N,T,P \rangle$ onde $N = \{S,A\}$, $T = \{0,1\}$ e $P = \{S \to 0SA, S \to 1, A \to 0AA, A \to 00\}$

$$f_1(s_1,s_2) = p(0SA/S)s_1s_2 + p(1/S)$$

$$f_2(s_1,s_2) = p(0AA/A)s_2^2 + p(00/A)$$

Dois níveis são equivalentes se eles contêm o mesmo número de símbolos não-terminais de cada tipo. Então a função de geração para o i-ésimo nível é definida como segue:

Definição 5.10 - A função de geração do i-ésimo nível $F_i(s_1,s_2,\ldots,s_K)$ é definida recursivamente como segue:

$$F_o(s_1,s_2,\ldots,s_K) = s_1$$

$$F_1(s_1,s_2,\ldots,s_K) = f_1(s_1,s_2,\ldots,s_K)$$

$$\vdots$$

$$F_i(s_1,s_2,\ldots,s_K) = F_{i-1}(f_1(s_1,s_2,\ldots,s_K),f_2(s_1,\ldots,s_K),\ldots,f_K(s_1,s_2,\ldots,s_K))$$

Exemplo 5.3 - Para gramática do exemplo 5.2, temos:

$$F_o(s_1,s_2) = s_1$$

$$F_1(s_1,s_2) = f_1(s_1,s_2) = p(0SA/S)s_1s_2 + p(1/S)$$

$$F_2(s_1,s_2) = p(0SA/S)\left[f_1(s_1,s_2)\right]\left[f_2(s_1,s_2)\right] + p(1/S)$$

$$= p^2(0SA/S)p(0AA/A)s_1s_2^2$$
$$+p^2(0SA/S)p(00/A)s_1s_2$$
$$+ p(0SA/S)p(1/S)p(0AA/A)s_2^2$$
$$+ p(0SA/S)p(1/S)p(00/A)$$
$$+ p(1/S)$$

Examinando o exemplo 5.3, podemos ver que $F_i(s_1,s_2,\dots,s_K)$ pode ser escrito como:

$$F_i(s_1,s_2,\dots,s_K) = G_i(s_1,s_2,\dots,s_K) + C_i$$

onde o polinômio $G_i(.)$ não contém nenhum termo constante. O termo C_i corresponde à probabilidade de todas as listas $x \in L(G)$, que pode ser derivada de i ou níveis menores. Mais detalhes em [19].

5.4 - Estudo da Consistência de Linguagens "Livre de Contexto"

Teorema 5.1 - A gramática livre de contexto com representação de probabilidade R é consistente se e somente se

$$\lim_{i\to\infty} C_i = 1$$ [19] [2] .

Definição 5.11 - O número esperado de ocorrência de símbolos não-terminais na produção ΓA_i é

$$e_{ij} = \left. \frac{\delta f_i(s_1,s_2,\dots, s_K)}{\delta s_j} \right/_{s_1,s_2,\dots,s_K = 1}$$ [2][4]

Definição 5.12 - A primeira matriz dos momentos E é definida como:

$$E = \left[e_{ij}\right] \qquad 1 \leq i,j \leq K$$

As raízes características ou auto-valores da matriz E serão indicados por $\rho_1, \rho_2, \ldots, \rho_K$ e tal que $|\rho_i| \leq |\rho_j|$ se $i < j$ [2][4][19].

Teorema 5.2 - Uma linguagem livre de contexto com representação de probabilidade R é consistente se $\rho_1 < 1$ e não-consistente se $\rho_1 > 1$ [2][4][19].

CAPÍTULO VI

ATRIBUIÇÃO DE PROBABILIDADE PARA GRAMÁTICAS INDICIAIS

Neste capítulo iremos estudar a atribuição de probabilidade às produções de gramáticas indiciais, como também a consistência das linguagens geradas por elas. Para isto tornou-se necessário a prévia transformação da linguagem indicial dada para uma sensível ao contexto equivalente. Uma vez estudada a consistência em linguagens indiciais, como estas são subconjunto próprio das linguagens sensíveis ao contexto, talvez seja possível extender as idéias aqui utilizadas para o estudo da consistência de linguagens sensíveis ao contexto às quais probabilidades forem associadas.

6.1 - Transformação da Gramática Indicial

Teorema 6.1 - Toda gramática indicial $G = \langle N,T,F,P,S\rangle$ pode ser transformada em uma sensível ao contexto equivalente $G' = \langle N',T,P',S\rangle$ onde o conjunto de produções P' é da forma $\alpha \to \beta$, onde $\beta \in (NF^* \cup T)^*$, $\alpha \in NF \cup N$ e $N' = NF \cup N$. Os flags dessa gramática sensível ao contexto (transformada de G) se distribuem sobre os não-terminais e não-terminais indiciados, como nas gramáticas indiciais.

Demonstração:

Seja $\langle G = N,T,F,P,S\rangle$ uma gramática indicial, onde:

a) $N = \{A_1,\ldots,A_n\}$ o conjunto finito não vazio dos não-termi nais.

b) $T = \{a_1, a_2, \ldots, a_e\}$ conjunto finito não vazio dos terminais

c) $F = \{f_1, \ldots, f_s\}$ onde cada f_i $1 \leq i \leq s$ é da forma $\left[(B_1,X_1),(B_2,X_2),\ldots,(B_K,X_K)\right]$ com $B_1,B_2,\ldots,B_K \in N$ e $X_1,X_2,\ldots,X_K \in (N \cup T)^*$, tal que $1 \leq K \leq n$ e $\{B_1,B_2,\ldots,B_K\} \subseteq N$.

d) P é um conjunto de pares ordenados $\left[A_j,\alpha_j\right]$ onde $A_j \in N$ e $\alpha_j \in (NF^* \cup T)^*$ com $1 \leq j \leq m \leq n$ ou $A_1 \rightarrow \alpha_1$, $A_2 \rightarrow \alpha_2, \ldots, A_m \rightarrow \alpha_m$.

Sabemos que o conjunto F é definido da seguinte forma: $F = \{f_1, \ldots, f_s\}$ onde cada f_i $1 \leq i \leq s$ é da forma $\left[B_1 \rightarrow X_1, \ldots, B_K \rightarrow X_K\right]$ com $B_1,\ldots,B_K \in N$ e $X_1,\ldots,X_K \in (N \cup T)^*$.

Isto quer dizer que a produção $B_i \rightarrow X_i$ só vai ser aplicada quando aparecer ao longo de uma derivação a seguinte situação: $\ldots B_i f_i \ldots$ e $(B_i \rightarrow X_i) \in f_i$, o $B_i f_i$ será substituído por X_i na derivação seguinte. Então pela própria definição de F podemos adicionar as produções de f_i a P fazendo $B_i f_i \rightarrow X_i$ onde $(B_i \rightarrow X_i) \in f_i$. Então a gramática G' fica da seguinte forma:

$G' = \langle N',T,P',S \rangle$ onde $N' = N \cup \{B_i f_i\}$ $i=1,\ldots,K$

$P' = P \cup \{B_i f_i \rightarrow X_i\}$ $i=1,\ldots,K$

Então, pela própria construção, G' é equivalente a G.

Teorema 6.2 - Toda gramática indicial $G = \langle N,T,F,P,S \rangle$ pode ser transformada em uma outra equivalente $G' = \langle N',T,F,P',S \rangle$ (a menos de possíveis engasgos), onde os não-terminais que ficam à esquer da das produções de P' não pertencem aos não-terminais que ficam à esquerda das produções dos flags de F.

Demonstração:

Seja $N = \{A_1, \ldots, A_n\}$ e $\{A_1, \ldots, A_K\}$ $K \leq n$ o conjunto dos não-terminais que estão à esquerda das produções de P e estão também à esquerda das produções dos flags de F. Se substituirmos A_1 por B_1, os A_2 por B_2 até A_k por B_K, e acrescentarmos as produções $B_1 \to A_1$, $B_2 \to A_2, \ldots, B_K \to A_K$ ao conjunto de produções de P, a linguagem gerada pela gramática G' vai ser a mesma gerada por G (a menos engasgo), isto porque se substituirmos qualquer B_i por A_i ao longo de uma derivação, o A_i tem um flag ou flags à sua direita e só tem uma opção, a opção de ir para o conjunto de flags, isto porque não existe nenhum A_i à esquerda das produções de P'.

Exemplo 6.1 - Seja $G = \langle \{S,A\}, \{c\}, \{f\}, P, S \rangle$, onde P contém as produções

$$\begin{cases} S \to Af \\ A \to A \end{cases} \quad \text{e onde} \quad f = [A \to c]$$

Então $G' = \langle \{S,A,B\}, \{c\}, \{f\}, P', S \rangle$ onde P' contém as produções:

$$\begin{cases} S \to Bf \\ B \to B \\ B \to A \end{cases} \quad \text{e onde}$$

$$f = [A \to c]$$

exemplo 6.2 - Seja $G = \langle \{S,A,B\}, \{0,1\}, \{f\}, P, S \rangle$ onde P contém as produções

$$\begin{cases} S \to Af \\ A \to AB \end{cases}$$ e onde

$f = \left[A \to 0,\ B \to 1 \right]$

Então, $G' = \langle \{S,A,B,C\}, \{0,1\}, \{f\}, P',S \rangle$ onde P' contém as produções

$$\begin{cases} S \to Cf \\ C \to CB \\ C \to A \end{cases}$$ e onde

$f = \left[A \to 0 ,\ B \to 1 \right]$

Exemplo 6.3 - Seja $G = \langle \{S,A\}, \{a\}, \{f\}, P,S \rangle$ onde P contém as produções

$$\begin{cases} S \to A \\ A \to Af \end{cases}$$ e onde

$f = \left[A \to a \right]$

Então $G' = \langle \{S,A,B\}, \{a\}, \{f\}, P',S \rangle$ onde P contém as produções

$$\begin{cases} S \to B \\ B \to Bf \\ B \to A \end{cases}$$ e onde

$f = \left[A \to c \right]$

Ambas as gramáticas G e G' dão engasgo. Para G, temos:

$$\begin{tikzcd}
 & & & & \mathrm{Afff} \\
 & & & \mathrm{Aff} \arrow[ur] \arrow[r] & \mathrm{af} \\
 & & \mathrm{Af} \arrow[ur] \arrow[dr] & & \text{engasgo} \arrow[u] \\
S \arrow[r] & A \arrow[ur] \arrow[dr] & & a & \\
 & & a & &
\end{tikzcd}$$

Para G', temos:

$$\begin{tikzcd}
S \arrow[r] & B \arrow[r] & \mathrm{Bf} \arrow[r] & \mathrm{Bff} \arrow[r] & \mathrm{Aff} \arrow[r] & \mathrm{af} \\
 & & & & & \text{engasgo} \arrow[u]
\end{tikzcd}$$

Definição 6.1 - Diz-se que em uma gramática indicial $G = \langle N,T,F,P,S \rangle$ P e F estão interligados quando o conjunto de flags de F está distribuído à direita das produções de P, e todos os flags são utilizados ao longo das derivações.

O problema que surge na atribuição de probabilidade em produções de gramáticas indiciais é a existência de dois conjuntos de produção P e F .

A aplicação do processo de Galton-Watson para processos de Ramificação[4], tem como imposição principal que: se estivermos em um nó de uma árvore (no caso derivação) e pudermos ir para vários nós dessa árvore, a soma de probabilidades de ir para todos os possíveis nós a partir deste nó é 1 (um).

No caso de derivação em gramáticas indiciais surge o seguinte problema. Se temos um nó com a lista Af ,tanto podemos ir de A para outra lista, como podemos ir de Af para alguma outra lista. O problema então é construir uma gramática equivalente, onde, quando surgir um Af, só podemos ir de Af para alguma lista ou de A para alguma lista, ou seja, A pode surgir à esquerda de alguma produção de P, ou A pode surgir à esquerda de alguma produção de f, não podendo portanto surgir nos dois casos ao mesmo tempo. Ou podemos transformar a gramática indicial para que fique nesta condição, como fizemos no teorema 6.2 .

Definição 6.2 - "Não-terminal ponte"são os não-terminais que servem de ligação para aplicação do conjunto de produções dos flags. Ou seja, são os não-terminais que figuram à direita das produções de P e são premissas das produções de algum flag de F. Nas gramáticas indiciais transformadas pelo teorema 6.2, os não-terminais ponte são os não-terminais que são premissas das produções dos flags ou os não-terminais que não são premissas das produções de P de G', onde G' é a gramática indicial transformada de G.

Exemplo 6.4 - Seja $G = \langle\{S,A,B\},\{a,b,c\},\{f,g\},P,S\rangle$ onde as produções contidas de P são:

$$\begin{cases} S \to aAfc \\ A \to aAgc \\ A \to B \end{cases}$$ e onde

$$f = \left[B \to b\right]$$

$$g = \left[B \rightarrow bB\right]$$

Temos somente um não-terminal ponte que é o não-terminal B .

Exemplo 6.5 - Seja $G = \langle \{S,A\},\{a\}, \{f\}, P,S \rangle$, onde as produções contidas em P são:

$$\begin{cases} S \rightarrow Af \\ A \rightarrow A \end{cases}$$ e onde

$$f = \left[A \rightarrow a\right]$$

Temos como não-terminal ponte o não-terminal A .

6.2 - Estudo da Consistência

Nesse ítem, nós iremos sugerir uma maneira de atribuir probabilidade às produções de uma gramática indicial, bem como estudar a consistência da linguagem gerada por essa gramática.

Para atribuirmos probabilidades às produções de uma gramática indicial transformada pelos teoremas 6.1 e 6.2, atribuímos probabilidades tanto às produções que tem não-terminal como premissa, como às produções que tem não-terminal indicial como premissa.

Seja $P = \Gamma A_1 \cup \Gamma A_2 \cup \ldots \cup \Gamma K$ a partição de P na classe de equivalência tal que duas produções estão na mesma classe se e somente se elas tem a mesma premissa. Para cada ΓA_j, definimos probabilidades para as regras de produção $A_j \rightarrow \alpha$ ($A_j \in NF \cup N$ e

$\alpha \in (NF^* \cup T)^*$ e denotamos por $\{p(\alpha/A_j)\}$ onde $\sum_{\Gamma A_j} p(\alpha/A_j) = 1$.

Então a atribuição de probabilidade às produções é dada equivalentemente a atribuição de probabilidade às produções de gramática livre de contexto proposta por Booth e Thompson [2], [19], [20] .

Exemplo 6.6 - Seja $G = \langle \{S,A\}, \{c,b\}, \{f\}, P, S \rangle$, onde P contém as produções

$$\begin{cases} S \rightarrow Af \\ A \rightarrow A \end{cases}$$ e onde

$$f = [A \rightarrow a \ , \ A \rightarrow b]$$

Pelo teorema 6.2, temos $G' = \langle \{S,A,B\}, \{a,b\}, \{f\}, P', S \rangle$ onde P' contém as produções

$$\begin{cases} S \rightarrow Bf \\ B \rightarrow B \\ B \rightarrow A \end{cases}$$ e onde

$$f = [A \rightarrow c \ , \ A \rightarrow b] \quad .$$

Pelo teorema 6.1, temos:

$S \rightarrow Bf$		1 $S \rightarrow Bf$
$B \rightarrow B$	a atribuição de	0.5 $B \rightarrow B$
$B \rightarrow A$	probabilidade	0.5 $B \rightarrow A$
$Af \rightarrow c$		0.5 $Af \rightarrow c$
$Af \rightarrow b$	será	0.5 $Af \rightarrow b$

Exemplo 6.7 - Seja $G = \langle \{S,A,B\},\{0,1\},\{f\},P,S \rangle$ onde P contém as produções

$$\begin{cases} S \to Af \\ A \to AB \end{cases} \quad \text{e onde}$$

$$f = \left[A \to 0 \ , \ \ B \to 1 \right]$$

Pelo teorema 6.2, temos $G' = \langle \{S,A,B,C\},\{0,1\}, \{f\}, P',S \rangle$, onde P' contém as produções

$$\begin{cases} S \to Cf \\ C \to CB \\ C \to A \end{cases} \quad \text{e onde}$$

$$f = \left[A \to 0, \ B \to 1 \right]$$

Pelo teorema 6.1, temos:

$S \to Cf$	a atribuição de probabilidade será	1	$S \to Cf$
$C \to CB$		0.5	$C \to CB$
$C \to A$		0.5	$C \to A$
$Af \to 0$		1	$Af \to 0$
$Bf \to 1$		1	$Bf \to 1$

Nas gramáticas livres de contexto, na aplicação do processo de Booth e Thompson [2][19] , para estudo das probabilidades, numerávamos os não-terminais para efeito de construção das funções f_i. Agora, para gramáticas indiciais, vamos numerar tanto os não-terminais como os não-terminais indiciados. Sendo que os não-terminais ponte são numerados condicionalmente da seguinte maneira: se o não-terminal ponte A_i estiver em $f_1, f_2, \ldots, f_n$ e $A_i f_1$ é numerado com 1 , $A_i f_2$ com 2, ..., e $A_i f_n$ com n, então A_i será numerado com $1 \vee 2 \vee 3 \vee \ldots \vee n$. Isto porque os não-terminais ponte por definição não pertencem ao conjunto de premissas de P, portanto eles só serão usados quando aparecer um flag ou conjunto de flags à sua direita.

As gramáticas apresentadas nos exemplos abaixo já estão na forma transformada pelos teoremas 6.1 e 6.2 .

Exemplo 6.8 - Seja $G = \langle \{S,A,B\}, \{a,b\}, \{f,g\}, P,S \rangle$, onde P contêm as seguintes produções

$S \rightarrow aAfc$		1	$S \rightarrow aAfc$
$A \rightarrow aAgc$	a atribuição de probabilidade será	0.5	$A \rightarrow aAgc$
$A \rightarrow B$		0.5	$A \rightarrow B$
$Bf \rightarrow b$		1	$Bf \rightarrow b$
$Bg \rightarrow bB$		1	$Bg \rightarrow bB$

$$\begin{matrix} 1 & 2 & 3 \vee 4 & 3 & 4 \\ \{ S, & A, & B, & Bf, & Bg \} \end{matrix}$$

$$f_1(s_1,s_2,s_3,s_4) = p(aAfc/S)s_2$$
$$f_2(s_1,s_2,s_3,s_4) = p(aAgc/A)s_2 + p(B/A)(s_3 \vee s_4)$$
$$f_3(s_1,s_2,s_3,s_4) = p(b/Bf)$$
$$f_4(s_1,s_2,s_3,s_4) = p(bB/Bg)(s_3 \vee s_4)$$

$$F_1(s_1,s_2,s_3,s_4) = s_1$$

$$F_2(s_1,s_2,s_3,s_4) = p(aAfc/S)s_2$$

$$F_3(s_1,s_2,s_3,s_4) = p(aAfc/S)p(aAgc/A)s_2 + p(aAfc/S)p(B/A)s_3$$

$$\begin{aligned} F_4(s_1,s_2,s_3,s_4) = {} & p(aAfc/S)p(aAgc/A)p(aAgc/A)s_2 + \\ & + p(aAfc/S)p(aAgc/A)p(B/A)s_4 + \\ & + \underbrace{p(aAfc/S)p(B/A)p(b/Bf)}_{1/2} \end{aligned}$$

$$\begin{aligned} F_5(s_1,s_2,s_3,s_4) = {} & p(aAfc/S)p(aAgc/A)p(aAgc/A)p(aAgc/A)s_2 + \\ & + p(aAfc/S)p(aAgc/A)p(aAgc/A)p(B/A)s_4 + \\ & + p(aAfc/S)p(aAgc/A)p(B/A)p(bB/Bg)s_3 + \\ & + \underbrace{p(aAfc/S)p(B/A)p(b/Bf)}_{1/2} \end{aligned}$$

$$\begin{aligned} F_6(s_1,s_2,s_3,s_4) = {} & p(aAfc/S)p(aAgc/A)p(aAgc/A)p(aAgc/A)s_2 + \\ & + p(aAfc/S)p(aAgc/A)p(aAgc/A)p(B/A)p(bB/Bg)s_4 + \\ & + \underbrace{p(aAfc/S)p(aAgc/A)p(B/A)p(bB/Bg)p(b/Bf)}_{1/4} \\ & + \underbrace{p(aAfc/S)p(B/A)p(b/Bf)}_{1/2} \end{aligned}$$

Então, $\lim_{i\to\infty} C_i = 1$, consequentemente, a linguagem $L(G)$ é consistente.

Exemplo 6.9 - Seja $G = \langle \{S,A,B,C,D\}, \{b,c\}, \{f\}, P,S \rangle$, onde P contém as produções seguintes:

$S \to Af$		1	$S \to Af$
$A \to BC$	atribuindo	1	$A \to BC$
$D \to b$	probabilidade temos	1	$D \to b$
$Bf \to D$		1	$Bf \to D$
$Cf \to c$		1	$Cf \to c$

$$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 4 & 5 \\ \{S, & A, & D, & Bf, & Cf, & B, & C\} \end{array}$$

$$f_1(s_1,s_2,s_3,s_4,s_5) = p(Af/S)s_2$$
$$f_2(s_1,s_2,s_3,s_4,s_5) = p(BC/S)s_4s_5$$
$$f_3(s_1,s_2,s_3,s_4,s_5) = p(b/D)$$
$$f_4(s_1,s_2,s_3,s_4,s_5) = p(D/Bf)s_3$$
$$f_5(s_1,s_2,s_3,s_4,s_5) = p(c/Cf)$$

$$F_1(s_1,s_2,s_3,s_4,s_5) = s_1$$
$$F_2(s_1,s_2,s_3,s_4,s_5) = p(Af/S)s_2$$
$$F_3(s_1,s_2,s_3,s_4,s_5) = p(Af/S)p(BC/A)s_4s_5$$
$$F_4(s_1,s_2,s_3,s_4,s_5) = p(Af/S)p(BC/A)p(D/Bf)s_3p(c/Cf)$$
$$F_5(s_1,s_2,s_3,s_4,s_5) = p(Af/S)p(BC/A)p(D/Bf)p(c/Cf)p(b/D)$$

Então, a linguagem gerada por G é consistente.

Exemplo 6.10 - Seja $G = \langle \{S,A,B,C,D\}, \{b,c\}, \{f\}, P,S \rangle$, onde P contém as produções seguintes:

$S \rightarrow Af$		1	$S \rightarrow Af$
$A \rightarrow BC$	atribuindo	1	$A \rightarrow BC$
$D \rightarrow b$	probabilidades	0.5	$D \rightarrow b$
$D \rightarrow DD$	temos	0.5	$D \rightarrow DD$
$Bf \rightarrow D$		1	$Bf \rightarrow D$
$Cf \rightarrow c$		1	$Cf \rightarrow c$

$$\begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 5 & 4 \\ \{S, & A, & D, & Bf, & Cf, & C, & B\} \end{array}$$

$$f_1(s_1,s_2,s_3,s_4,s_5) = p(Af/S)s_2$$
$$f_2(s_1,s_2,s_3,s_4,s_5) = p(BC/A)s_4s_5$$
$$f_3(s_1,s_2,s_3,s_4,s_5) = p(b/D)+p(DD/D)s_3^2$$
$$f_4(s_1,s_2,s_3,s_4,s_5) = p(D/Bf)s_3$$
$$f_5(s_1,s_2,s_3,s_4,s_5) = p(c/Cf)$$

$$F_1(s_1,s_2,s_3,s_4,s_5,) = s_1$$
$$F_2(s_1,s_2,s_3,s_4,s_5) = p(Af/S)s_2$$
$$F_3(s_1,s_2,s_3,s_4,s_5) = p(Af/S)p(BC/A)s_4s_5$$
$$F_4(s_1,s_2,s_3,s_4,s_5) = p(Af/S)p(BC/A)p(D/Bf)s_3p(c/Cf)$$
$$F_5(s_1,s_2,s_3,s_4,s_5) = \underbrace{p(Af/S)p(BC/A)p(D/Bf)p(c/Cf)p(b/D)}_{1/4}$$
$$+p(Af/S)p(BC/A)p(D/Bf)p(c/Cf)p(DD/D)s_3^2$$

$$F_6(s_1,s_2,s_3,s_4,s_5) = p(Af/S)p(BC/A)p(D/Bf)p(c/Cf)p(b/D) +p(Af/S)p(BC/A)p(D/Bf)p(c/Cf) p(DD/D)\left[p(b/D)+p(DD/D)s_3^2\right]$$

Vê-se facilmente que a linguagem gerada por G é inconsistente.

Exemplo 6.11 - Seja $G = \langle \{S,A,B\}, \{a\}, \{f\}, P,S \rangle$, onde P contêm as produções:

$$\begin{cases} S \rightarrow ABf \\ B \rightarrow a \\ Af \rightarrow a \end{cases}$$

Como Bf aparece à direita das produções e $B \rightarrow a$, então temos que modificar a gramática para aplicar o processo.

Seja $G' = \langle \{S,A,B,C\}, \{a\}, \{f\}, P',S \rangle$, onde P' contém as produções:

$S \rightarrow ACf$		1	$S \rightarrow ACf$
$C \rightarrow B$	atribuindo probabilidades temos	1/2	$C \rightarrow B$
$C \rightarrow a$		1/2	$C \rightarrow a$
$Af \rightarrow a$		1	$Af \rightarrow a$

$$\begin{matrix} 1 & 2 & 3 & 4 & 4 \\ \{S, & C, & B, & Af, & A\} \end{matrix}$$

$$f_1(s_1,s_2,s_3,s_4) = p(ACf/S)s_4s_2$$
$$f_2(s_1,s_2,s_3,s_4) = p(B/C)s_3 + p(a/C)$$
$$f_3(s_1,s_2,s_3,s_4) = p(a/Af)$$

$$F_1(s_1,s_2,s_3,s_4) = s_1$$
$$F_2(s_1,s_2,s_3,s_4) = p(ACf/S)s_4s_2$$
$$F_3(s_1,s_2,s_3,s_4) = p(ACf/S)s_4\left[p(B/C)s_3 + p(a/C)\right]$$

Podemos ver que dá inconsistência na linguagem.

Se fizermos a produção de G

$$\begin{cases} A \rightarrow AfB \\ B \rightarrow a \\ Af \rightarrow a \end{cases}$$

a linguagem será consistente.

Exemplo 6.12 - Seja $G = \langle \{S,A,B,C\}, \{a,b\}, \{f\}, P,S \rangle$, onde P contém as produções:

$$\begin{cases} S \rightarrow ABf \\ B \rightarrow a \\ A \rightarrow c \\ Cf \rightarrow b \end{cases}$$

Como na gramática do exemplo anterior, temos que modificar G para aplicarmos o processo.

Então, $G = \langle \{S,A,B,C,D\}, \{a,b\}, \{f\}, P',S \rangle$, onde P' contém as produções:

$$
\begin{array}{lcl}
S & \rightarrow & ADf \\
D & \rightarrow & B \\
D & \rightarrow & a \\
A & \rightarrow & C \\
Cf & \rightarrow & b
\end{array}
\qquad \text{atribuindo probabilidades temos} \qquad
\begin{array}{rlcl}
1 & S & \rightarrow & ADf \\
1/2 & D & \rightarrow & B \\
1/2 & D & \rightarrow & a \\
1 & A & \rightarrow & C \\
1 & Cf & \rightarrow & b
\end{array}
$$

$$
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 4 & 5 \\
\{S, & A, & D, & C, & Cf, & B\}
\end{array}
$$

$$f_1(s_1,s_2,s_3,s_4,s_5) = p(ADf/S)s_2s_3$$
$$f_2(s_1,s_2,s_3,s_4,s_5) = p(B/D)s_5 + p(a/D)$$
$$f_3(s_1,s_2,s_3,s_4,s_5) = p(C/A)s_4$$
$$f_4(s_1,s_2,s_3,s_4,s_5) = p(b/Cf)$$

$$F_1(s_1,s_2,s_3,s_4,s_5) = s_1$$
$$F_2(s_1,s_2,s_3,s_4,s_5) = p(ADf/S)s_2s_3$$
$$F_3(s_1,s_2,s_3,s_4,s_5) = p(ADf/S)\left[p(B/D)s_5 + p(a/A)\right]p(C/A)s_4$$

Podemos ver facilmente que a linguagem gerada por G' é inconsistente.

Exemplo 6.13 - Seja $G = \langle\{S,B,A\}, \{c\}, \{f\}, P, S\rangle$, em que P contém:

$$
\begin{array}{lcl}
S & \rightarrow & Bf \\
B & \rightarrow & B \\
B & \rightarrow & A \\
Af & \rightarrow & c
\end{array}
\qquad \text{atribuindo probabilidades temos} \qquad
\begin{array}{rlcl}
1 & S & \rightarrow & Af \\
1/2 & B & \rightarrow & B \\
1/2 & B & \rightarrow & A \\
1 & Af & \rightarrow & c
\end{array}
$$

$$\begin{array}{cccc} 1 & 2 & 3 & 3 \\ \{S, & B, & Af, & A\} \end{array}$$

$$f_1(s_1,s_2,s_3) = p(Bf/S)s_2$$
$$f_2(s_1,s_2,s_3) = p(B/B)s_2+p(A/B)s_3$$
$$f_3(s_1,s_2,s_3) = p(c/Af)$$

$$F_1(s_1,s_2,s_3) = s_1$$
$$F_2(s_1,s_2,s_3) = p(Bf/S)s_2$$
$$F_3(s_1,s_2,s_3) = p(Bf/S)\left[p(B/B)s_2+p(A/B)s_3\right]$$
$$= p(Bf/S)p(B/B)s_2+p(Bf/S)p(A/B)s_3$$
$$F_4(s_1,s_2,s_3) = p(Bf/S)p(B/B)\left[p(B/B)s_2+p(A/B)s_3\right]+p(Bf/S)p(A/B)\,p(c/Af) =$$
$$= p(Bf/S)p(B/B)p(B/B)s_2+p(Bf/S)p(B/B)p(A/B)s_3+$$
$$\underbrace{+p(Bf/S)p(A/B)p(c/Af)}_{1/2}$$

$$F_5(s_1,s_2,s_3,s_4) = p(Bf/S)p(B/B)p(B/B)p(B/B)s_2 +$$
$$+p(Bf/S)p(B/B)p(B/B)p(A/B)s_3 +$$
$$\underbrace{+p(Bf/S)p(B/B)p(A/B)p(c/Af)}_{1/4}$$
$$\underbrace{+p(Bf/S)p(A/B)p(c/Af)}_{1/2}$$

Podemos ver facilmente que a linguagem gerada por G é consistente.

Exemplo 6.14 - Seja $G = \langle \{S,A,B\}, \{0,1\}, \{f\}, P,S \rangle$, onde P contém as produções:

$$\begin{cases} S \rightarrow Af \\ A \rightarrow AB \quad \text{e onde} \end{cases}$$

$$f = [A \rightarrow 0,\ B \rightarrow 1]$$

que é equivalente a :

$S \rightarrow Cf$		$1 \quad S \rightarrow Cf$
$C \rightarrow CB$	atribuindo	$1/2 \quad C \rightarrow CB$
$C \rightarrow A$	probabilidades	$1/2 \quad C \rightarrow A$
$Af \rightarrow 0$	temos	$1 \quad Af \rightarrow 0$
$Bf \rightarrow 1$		$1 \quad Bf \rightarrow 1$

$$\{\overset{1}{S}, \overset{2}{C}, \overset{3}{Af}, \overset{4}{Bf}, \overset{3}{A}, \overset{4}{B}\}$$

$$f_1(s_1,s_2,s_3,s_4) = p(Cf/S)s_2$$
$$f_2(s_1,s_2,s_3,s_4) = p(CB/C)s_2s_4+p(A/C)s_3$$
$$f_3(s_1,s_2,s_3,s_4) = p(0/Af)$$
$$f_4(s_1,s_2,s_3,s_4) = p(1/Bf)$$

$$F_1(s_1,s_2,s_3,s_4) = s_1$$
$$F_2(s_1,s_2,s_3,s_4) = p(Cf/S)s_2$$
$$F_3(s_1,s_2,s_3,s_4) = p(Cf/S)p(CB/C)s_2s_4+p(Cf/S)p(A/C)s_3$$
$$F_4(s_1,s_2,s_3,s_4) = p(Cf/S)p(CB/C)s_2s_4p(1/Bf)+$$
$$+p(Cf/S)p(CB/C)p(A/C)s_3p(1/Bf) +$$
$$\underbrace{+p(Cf/S)p(A/C)p(0/Af)}_{1/2}$$

$$\begin{aligned} F_5(s_1,s_2,s_3,s_4) = {} & p(Cf/S)p(CB/C)p(CB/C)\Big[p(CB/C)s_2s_4 + \\ & +p(A/C)s_3\Big]\ p(1/Bf)p(1/Bf)\ + \\ & \underbrace{+p(Cf/S)p(CB/C)p(A/C)p(0/Af)p(1/Bf)}_{1/4} \end{aligned}$$

$$\underbrace{p(Cf/S)p(A/C)p(0/Af)}_{1/2}$$

Então, L(G) é consistente.

CAPÍTULO VII

GERAÇÃO DE PALAVRAS DE UMA LINGUAGEM INDICIAL PROBABILÍSTICA POR MEIO DO COMPUTADOR

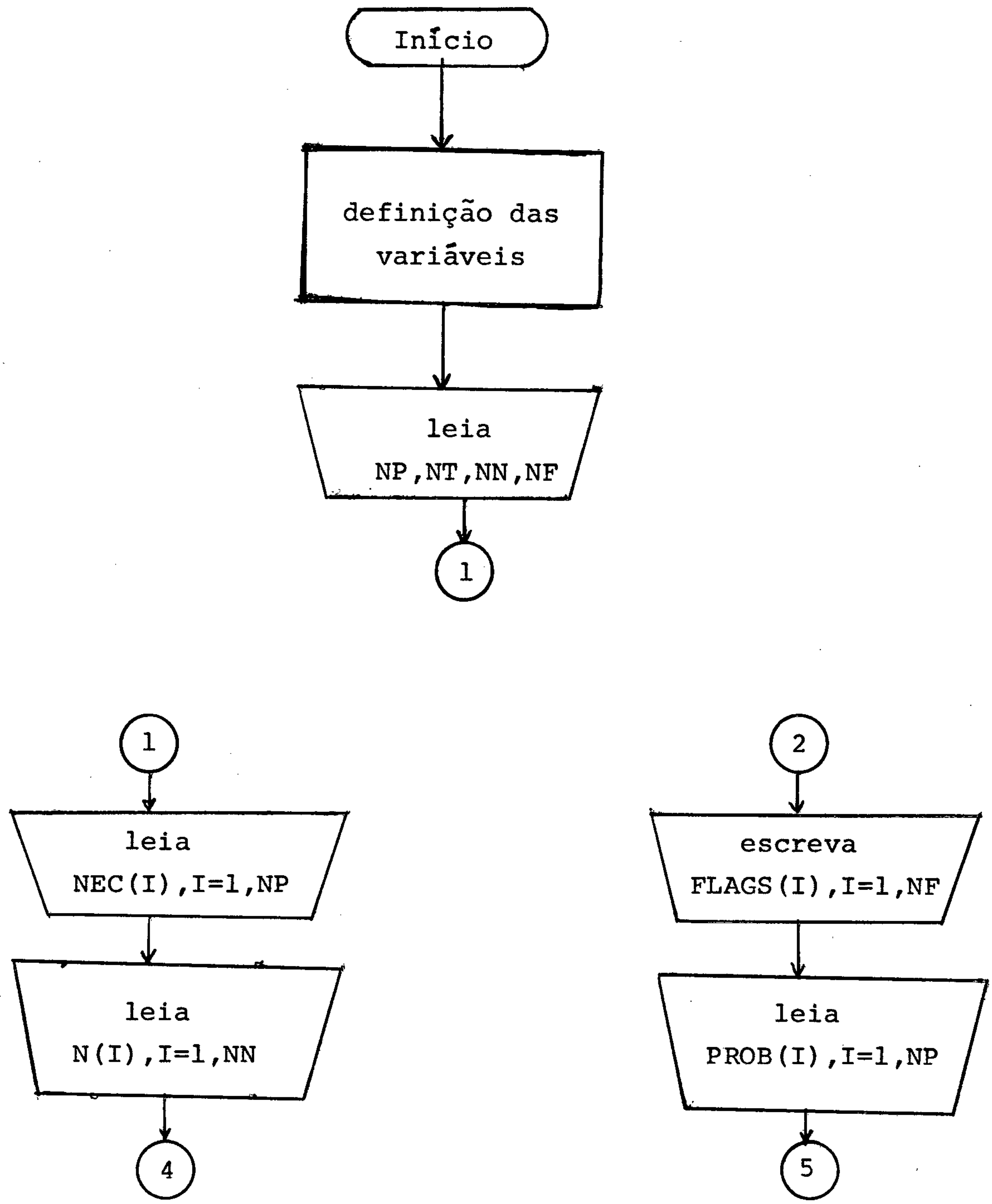
Neste capítulo descreveremos um programa que gera palavras de uma linguagem indicial probabilística, gráficos das distribuições probabilísticas das palavras, e são testadas várias gramáticas indiciais probabilísticas, como também é feito o estudo dessa distribuição.

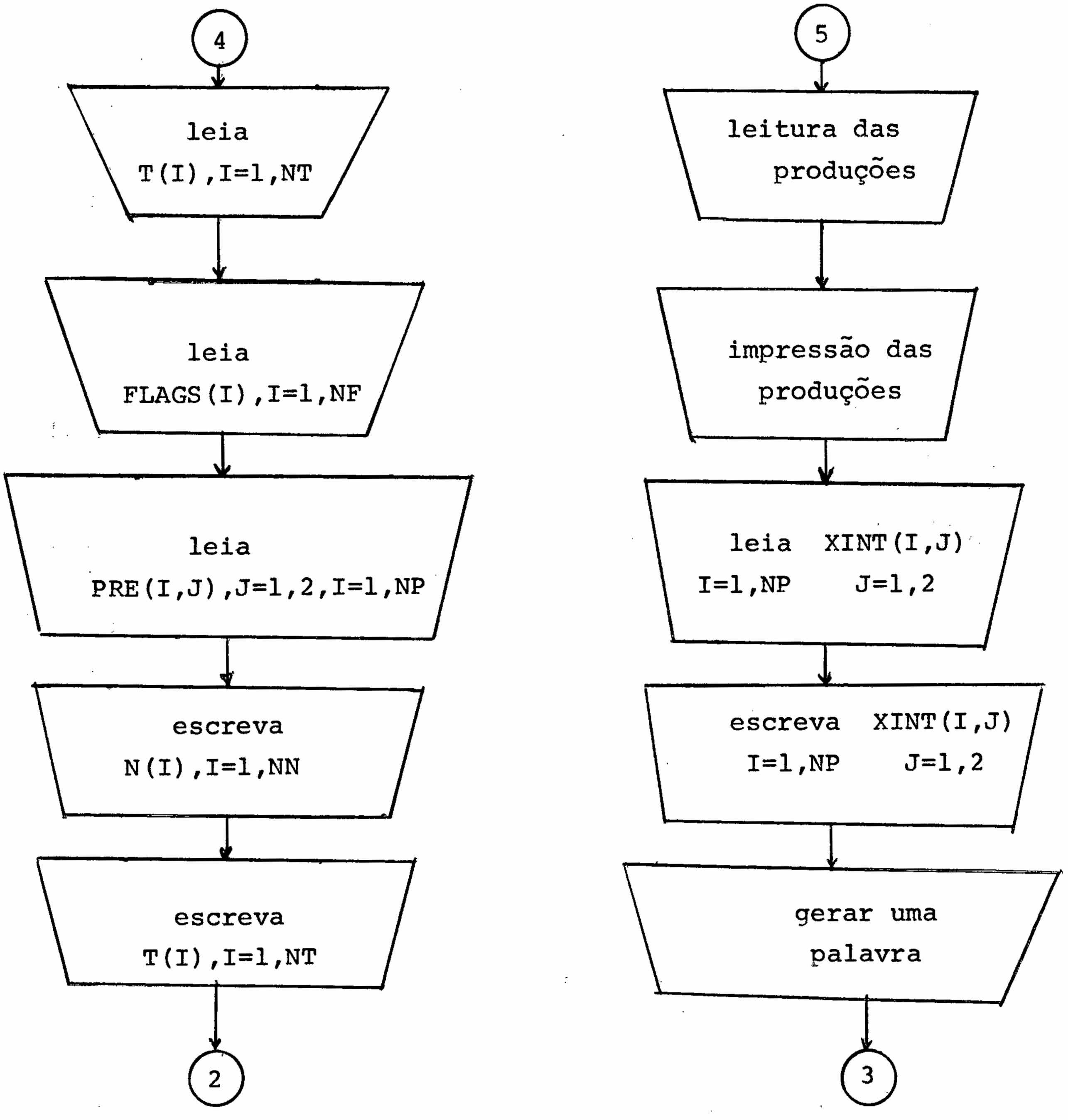
7.1 - Fluxograma simplificado do programa que gera palavras de uma linguagem indicial probabilística a partir de uma gramática indicial probabilística transformada

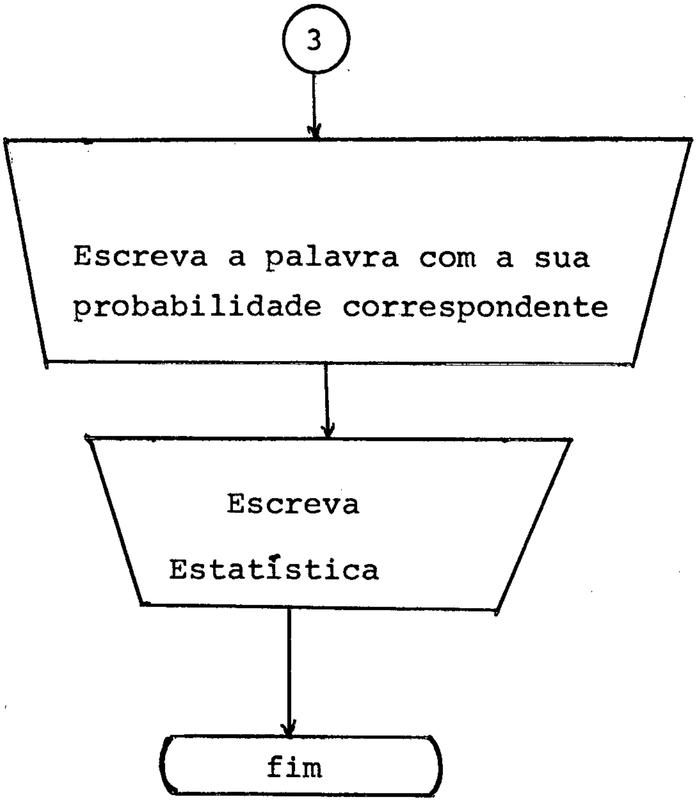
As definições das variáveis estão no programa do apêndice A, juntamente com várias gramáticas que foram testadas. Os não-terminais são seguidos do sinal "+" e os terminais, do sinal "_" .

As premissas das produções têm dois campos: o primeiro é reservado para um não-terminal e o segundo para um flag ou um branco. A derivação considerada no programa é a derivação mais à esquerda. O programa foi rodado no computador IBM - 1130 , e o tempo de processamento é muito variável, dependendo da gramática. Foram rodados vários exemplos, entretanto não foi considerada a lista gem de todas as palavras, somente a gramática, uma amostra das pa - lavras e a estatística para efeito de estudo.

Figura 7.1




4
leia
T(I),I=1,NT
leia
FLAGS(I),I=1,NF
leia
PRE(I,J),J=1,2,I=1,NP
escreva
N(I),I=1,NN
escreva
T(I),I=1,NT
2
5
leitura das
produções
impressão das
produções
leia XINT(I,J)
I=1,NP J=1,2
escreva XINT(I,J)
I=1,NP J=1,2
gerar uma
palavra
3



7.2 - <u>Estudo estatístico dos resultados obtidos no programa</u>

Os gráficos do apêndice B foram construídos com o auxílio do computador IBM - 1130 . Verificamos pelos gráficos obtidos que as linguagens indiciais probabilísticas geradas obedecem a lei de Zipf [21], ou seja, quaisquer que sejam as atribuições das probabilidades às produções, as frequências relativas observadas foram tais que a distribuição correspondente (se ordenarmos as palavras por ordem de frequência) aproximou-se da distribuição exponencial.

CAPÍTULO VIII

CONCLUSÕES

É estimado para um futuro próximo que os compiladores irão corrigir erros da linguagem que por transmissão sofreu erros, ou erro do próprio programador. Certamente o problema de semântica não será resolvido somente por atribuirmos probabilidades nas produções de uma gramática de uma maneira adequada, mas provavelmente, a distribuição probabilística irá ajudar na resolução desse problema.

Já foram definidas classes de linguagens maiores do que as classes de linguagens livres de contexto. Podemos citar como exemplo as gramáticas programadas [6]. Partes dessas classes foram construídas devido à insuficiência das gramáticas livres de contexto para especificar a estrutura sintática encontrada em muitos algorítmos de linguagens de programação dos nossos dias, tal como Algol,por exemplo [18].

Como as linguagens indiciais abrangem as livres de contexto e parte das sensíveis ao contexto, é possível uma linguagem mais poderosa, melhorando os sistemas de comunicação nos quais a fonte é gramatical.

Como as linguagens indiciais são incluídas propriamente nas sensíveis ao contexto, é provável que seja possível estudar a consistência de um conjunto de linguagens sensíveis ao contexto, maior do que o abrangido pelas linguagens indiciais, gerando portanto, um tema para pesquisa.

Outro tema para pesquisa seria a decodificação compacta das linguagens indiciais.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Booth, Taylor L. , "Sequential Machines and Automata Theory" , Johm Wiley and Sons, Inc. 1967 .

2. Booth, Taylor L. , "Probabilistic Representation of Formal Languages", IEEE Conference Record of the Tenth Annual Symposium on Switching and AutomataTheory, 1969 .

3. Ellis, Clarence A., "Probabilistic Languages and Automata" , Doctoral Thesis, Department of Computer Science, University of Illinois, 1969 .

4. Harris, T.E. , "The Theory of Branching Processes", Springer-Verley, Berlin, 1963.

5. Hopcroft, John E. and Ullman, J.D. , "Formal Languages and their Relation to Automata", Addison-Wesley Publ.Co., 1969.

6. Rosenkrantz, D.J. , "Programmed Grammars - A New Device for Generating Formal Languages" , Columbia University, 1967.

7. Aho, A.V. , "Indexed Grammars and Extension of Context Free Grammars", Austin, Texas, 1967.

8. Thompson, R. A. , "Compact Incoding of Probabilistic Languages" Underwater Sound Laboratory Research Project - Report USL,1971.

9. Ginsburgs, S. , "The Mathematical Theory of Context-Free Languages", McGraw-Hill, New York.

10 . Souza, C.R. , "Probabilities in Context-Free Programmed Grammars" , The Aloha System Tech. Report A 70-2, Univ. of Hawaii, Honolulu, 1970.

11 . Souza, C.R. , "Discrete and Stochastic Models and Formal Languages" , Notre Dame University.

12 . Souza, C.R. and Velasco,F.R. , "Sequential Syntactical Decoding, 1973 International Information Theory Symposium, Ashkelon, Israel, Junho/1973.

13 . Rabin, M.O., "Probabilistic Automata", Information and Control Vol.6, 1963.

14 . Zadeh, L.A. , "Fuzzy Set" , Information and Control, Vol.8, Junho/1965.

15 . Paz,A. , "Some Aspects of Probabilistic Automata", Information and Control, Vol.8, 1966.

16 . Fu, K.S. and L. , "Automata Games, Stochastic Automata and Formal Languages", Purdue Report 69-1 , Janeiro/1967.

17 . Chomsky,N. , "Three Models for the Description of Languages", IRE Transaction on Information Theory, Vol.IT-2, Setembro/ / 1956.

18 . Chomsky,N. , "Formal Properties of Grammars", Handbook of Mathematical Psychology, Vol.II, R.Luce, R. Bush and E. Galands (Eds) John Wiley and Sons, Inc., New York, 1963.

19 . Booth , Taylor L. and Thompson Richard A. , "Applying Probability Measure to Abstract Languages" - IEEE Transaction on Computers, Vol. C.22, Nº 5, Maio/1973.

20 . Thompson Richard A. , "Determination of Probabilistic Grammars for Functionally Specified Probability" - Measure Languages . (A ser publicado no"IEEE Transaction on Computers").

21 . Mandelbrot, B. , "An Informational Theory of the Statistical Structure of Languages" , Laboratoires d'Electronique et de Physique Appliquées, Paris.

22 . Arbib,M.A. - "Theories of Abstract Automata", Prentice Hall, 1969.

23 . Nelson , "Introduction to Automata" , Case Institute of Technology, Cleveland, Ohio, John Wiley and Sons, New York, London, 1968 .

24 . Minsky , "Computation Finite and Infinite Machines" - Prentice Hall, 1967.

25 . Harrison Michael, "Introduction to Automata Theory"

26 . McKay Donald M , "Information and Meaning" , The M.I.T. Press , Cambridge, Massachusets, and London, 1969.

27 . Garvin, Paul L. , "Natural Language and the Computer" , McGraw -Hill Book Company, 1963.

A P Ê N D I C E   A

```
PAGE    1

// JOB T

LOG DRIVE    CART SPEC    CART AVAIL  PHY DRIVE
  0000         0008         0008        0000

V2 M10   ACTUAL 16K  CONFIG 16K

// FOR
*LIST SOURCE PROGRAM
*EXTENDED PRECISION
*ONE WORD INTEGERS
      SUBROUTINE ORI(IX,IY,YFL)
      IY=IX*899
      IF(IY)5,6,6
 5    IY=IY+32767+1
 6    YFL=IY
      YFL=YFL/32767.
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 EXTENDED PRECISION

CORE REQUIREMENTS FOR ORI
 COMMON      0  VARIABLES      0  PROGRAM     54
```

```
PAGE   2

RELATIVE ENTRY POINT ADDRESS IS 0006 (HEX)

END OF COMPILATION

// DUP

*STORE       WS  UA  ORI
CART ID 0008   DB ADDR  3DC1   DB CNT   0005

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
*EXTENDED PRECISION
*IOCS(CARD,1132PRINTER)
C
C***   PROGRAM QUE GERA AS PALAVRAS DE UMA LINGUAGEM INDICIAL
C***   OBJETIVO - TESE - DATA 10/04/73
C***    AUTOR - ORION DE OLIVEIRA SILVA
C
C***   DEFINICAO DAS VARIAVEIS
C
C***   PRE - PREMISSA DAS PRODUCOES
C***   CON - CONSEQUENCIA DAS PRODUCOES
C***   T - SIMBOLOS TERMINAIS
C***   FLAGS - CONJUNTO DE FLAGS
C***   NT - NUMERO DE SIMBOLOS TERMINAIS
C***   NF - NUMERO DE FLAGS
C***   NP - NUMERO DE PRODUCOES
```

```
PAGE    3

C***    N - SIMBOLOS NAO TERMINAL
C***    NN - NUMERO DE SIMBOLOS NAO TERMINAL
C***    NEC - NUMERO DE ELEMENTOS DA CONSEGUENCIA DE CADA PRODUCAO
C***    PROB - PROBABILIDADE DAS PRODUCOES
C***    XINT - INTERVALO DE PROBABILIDADE DE CADA PRODUCAO
C***    AUX - LISTA AUXILIAR PARA AS FORMAS SENTENCIAIS
C***    LISTA - VARIAVEL ONDE SERAO GERADAS AS PALAVRAS
C***    NECA - VARIAVEL AUXILIAR PARA DESTRIBUICAO DOS FLAGS
C***
C
C***    OS NAO TERMINAIS SAO GEGUIDOS SO SINAL '' + ''
C***    OS TERMINAIS SAO SEGUIDOS DO SINAL '' - ''
C
      INTEGER   PRE(30,2),CON(30,20),T(80),N(30),FLAGS(10),NEC(30),AUX(5
     100),LISTA(500),BLK
      DIMENSION PROB(30),XINT(30,2),YYY(1000),ITA(50)
      DATA MA/'+'/,BLK/' '/
C
C***    VARIAVEL INDICIAL DA SUBROTINA ORI
C
      IX=3
C
C***    LEITURA DAS VARIAVEIS
C
      READ(2,10) NP,NT,NN,NF
      READ(2,10) (NEC(I),I=1,NP)
 10   FORMAT(20I2)
      READ(2,20) (N(I),I=1,NN)
```

```
PAGE    4

      READ(2,20) (T(I),I=1,NT)
      READ(2,20) (FLAGS(I),I=1,NF)
      READ(2,20) ((PRE(I,J),J=1,2),I=1,NP)
 20   FORMAT(80A1)
C
C***   IMPRESSAO DA GRAMATICA
C
      WRITE(3,30)
 30   FORMAT(/,10X,'GRAMATICA G',/)
      WRITE(3,40) (N(I),I=1,NN)
 40   FORMAT(10X,'NAO TERMINAIS = ',80A1)
      WRITE(3,50) (T(I),I=1,NT)
 50   FORMAT(10X,'TERMINAIS =',40A1)
      WRITE(3,60) (FLAGS(I),I=1,NF)
 60   FORMAT(10X,'FLAGS =',40A1)
C
C***   IMPRESSAO DAS PRODUCOES
C
      WRITE(3,70)
 70   FORMAT(/,10X,'CONJUNTO DAS PRODUCOES',/)
      READ(2,75) (PROB(I),I=1,NP)
 75   FORMAT(10F5.3)
      DO 80 I=1,NP
      NJ=NEC(I)
      READ(2,20) (CON(I,J),J=1,NJ)
 80   WRITE(3,90) PROB(I),I,(PRE(I,J),J=1,2),(CON(I,J),J=1,NJ)
 90   FORMAT(10X,F7.5,1X,I2,1X,2A1,'-----* ',40A1)
C
```

```
PAGE   5

C***    LEITURA E IMPRESSAO DO INTERVALO DAS PRODUCOES
C
      WRITE(3,120)
  120 FORMAT(/,10X,'INTERVALO DE APLICACAO DAS PRODUCOES',/)
      DO 140 I=1,NP
      READ(2,130) (XINT(I,J),J=1,2)
  130 FORMAT(2F7.5)
  140 WRITE(3,150) I,(XINT(I,J),J=1,2)
  150 FORMAT(10X,'PRODUCAO',I3,2F9.5)
      MMM=1000
      WRITE(3,160)
  160 FORMAT(/,10X,'GERACAO DAS PALAVRAS',/)
      DO 290 I=1,MMM
      NJ=NEC(1)
      DO 170 I1=1,NJ
      LISTA(I1)=CON(1,I1)
  170 AUX(I1)=LISTA(I1)
      P=1.
  180 NJ1=NJ-1
      CALL ORI(IX,IA,X)
      IX=IA
      DO 260 J=1,NJ1
      IF(LISTA(J+1)-MA)260,185,260
  185 DO 250 K=1,NP
      ITES=1
      ITEST=1
      IF(LISTA(J)-PRE(K,1))250,2001,250
 2001 IF(X-XINT(K,1))250,250,2002
```

```
PAGE    6

2002  IF(X-XINT(K,2))2003,250,250
2003  IF(PRE(K,2)-BLK)2004,210,2004
2004  I3=J+2
      DO 186 I4=1,NF
      IF(LISTA(I3)-FLAGS(I4))186,2005,186
2005  IF(LISTA(I3)-PRE(K,2))250,187,250
 186  CONTINUE
      GO TO 251
 187  ITEST=0
      KK=NEC(K)
      NK=0
      NI=J+KK-1
      DO 190 I1=J,NI
      NK=NK+1
 190  LISTA(I1)=CON(K,NK)
      KK=J+2
      DO 200 I1=KK,NJ
      IF(I1-I3)3000,200,3000
3000  NI=NI+1
      LISTA(NI)=AUX(I1)
 200  CONTINUE
      P=P*PROB(K)
      NJ=NI
      DO 201 I1=1,NJ
 201  AUX(I1)=LISTA(I1)
      GO TO 300
 210  KK=NEC(K)
      NK=0
```

PAGE 7

```
      NI=J+KK-1
      DO 230 I2=J,NI
      NK=NK+1
 230  LISTA(I2)=CON(K,NK)
      KK=J+2
      DO 240 I2=KK,NJ
      NI=NI+1
 240  LISTA(NI)=AUX(I2)
      P=P*PROB(K)
      NJ=NI
      DO 246 I2=1,NJ
 246  AUX(I2)=LISTA(I2)
 300  N1=0
      M2=0
      DO 241 I2=1,NJ
      IF(AUX(I2)-MA)960,2006,960
2006  M2=M2+1
 960  DO 241 I4=1,NF
      IF(AUX(I2)-FLAGS(I4))241,2007,241
2007  IF(N1)2008,944,2008
2008  IF((II1+1)-I2)292,944,292
 944  II1=I2
      N1=N1+1
      ITA(N1)=AUX(I2)
 241  CONTINUE
      IF(N1)292,245,292
 292  I2=II1
      I4=0
```

```
PAGE   8

 254   ITE=1
 244   I4=I4+1
       IF(I4-(I2-N1-1))2009,255,255
2009   IF(AUX(I4)-MA)244,2010,244
2010   I5=I4+1
       I7=I4
       I4=I4+N1
       DO 401 II9=1,N1
       I7=I7+1
 401   LISTA(I7)=ITA(II9)
       ITE=0
       DO 243 I6=I5,NJ
       I7=I7+1
 243   LISTA(I7)=AUX(I6)
       NJ=I7
       I2=I2+N1
       DO 247 I9=1,NJ
 247   AUX(I9)=LISTA(I9)
       IF(M2-1)255,2011,255
2011   I5=I2+1
       I7=I2-1
       DO 900 I9=I5,NJ
       I7=I7+1
 900   LISTA(I7)=AUX(I9)
       NJ=I7
       DO 901 I9=1,NJ
 901   AUX(I9)=LISTA(I9)
 255   IF(ITE)245,254,245
```

```
 245  ITEST=0
      GO TO 180
 250  CONTINUE
      IF(ITEST-1)180,251,180
 251  WRITE(3,252)
 252  FORMAT(10X,'ENGASGO',/)
      GO TO 1000
 260  CONTINUE
      YYY(I)=P
      IF(I-32)3006,3006,290
 3006 WRITE(3,205) P
 205  FORMAT(10X,'P=',F8.6)
      DO 1002 IAM=2,NJ,2
 1002 LISTA(IAM)=BLK
      WRITE(3,202) (LISTA(I3),I3=1,NJ)
 202  FORMAT(10X,80A1)
 290  CONTINUE
      WRITE(3,410)
 410  FORMAT(//,10X,'E S T A T I S T I C A',//)
      WRITE(3,420) MMM
 420  FORMAT(10X,'AMOSTRA DE ',I5,1X,' LISTAS',/)
      MMM1=MMM-1
      DO 440 I=1,MMM1
      IF(YYY(I))2012,440,2012
2012  IL=I+1
      ICONT=1
      DO 430 J=IL,MMM
      IF(YYY(I)-YYY(J))430,2013,430
```

```
PAGE  10

2013  ICONT=ICONT+1
      YYY(J)=0.
 430  CONTINUE
      WRITE(3,450) ICONT,YYY(I)
 450  FORMAT(10X,I5,1X,'LISTAS COM PROBABILIDADE',1X,F8.6)
 440  CONTINUE
 1000 CALL EXIT
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 EXTENDED PRECISION
 IOCS

CORE REQUIREMENTS FOR
 COMMON      0  VARIABLES   5176  PROGRAM   1732

END OF COMPILATION

// XEQ
```

```
GRAMATICA G

NAO TERMINAIS = SAB
TERMINAIS =ABC
FLAGS =FG

CONJUNTO DAS PRODUCOES

1.00000  1 S -----* A-A+FC-
0.50000  2 A -----* A-A+GC-
0.50000  3 A -----* B+
1.00000  4 BF-----* B-
1.00000  5 BG-----* B-B+

INTERVALO DE APLICACAO DAS PRODUCOES

PRODUCAO  1  0.00000  1.00000
PRODUCAO  2  0.00000  0.50000
PRODUCAO  3  0.50000  1.00000
PRODUCAO  4  0.00000  1.00000
PRODUCAO  5  0.00000  1.00000

GERACAO DAS PALAVRAS

P=0.250000
A A B B C C
P=0.125000
A A A B B B C C C
```

```
P=0.500000
A B C
P=0.250000
A A B B C C
P=0.500000
A B C
P=0.500000
A B C
P=0.500000
A B C
P=0.062500
A A A A B B B B C C C C
P=0.250000
A A B B C C
P=0.500000
A B C
P=0.500000
A B C
P=0.500000
A B C
P=0.250000
A A B B C C
P=0.250000
A A B B C C
P=0.500000
A B C
P=0.500000
A B C
P=0.500000
A B C
```

```
P=0.500000
A B C
P=0.125000
A A A B B B C C C
P=0.500000
A B C
P=0.500000
A B C
P=0.250000
A A B B C C
P=0.500000
A B C
P=0.500000
A B C
P=0.250000
A A B B C C
P=0.125000
A A A B B B C C C
P=0.125000
A A A B B B C C C
P=0.250000
A A B B C C
P=0.250000
A A B B C C
P=0.031250
A A A A A B B B B B C C C C C
P=0.125000
A A A B B B C C C
P=0.007812
A A A A A A A B B B B B B B C C C C C C C
```

E S T A T I S T I C A

AMOSTRA DE 1000 LISTAS

```
244 LISTAS COM PROBABILIDADE 0.250000
105 LISTAS COM PROBABILIDADE 0.125000
537 LISTAS COM PROBABILIDADE 0.500000
 52 LISTAS COM PROBABILIDADE 0.062500
 29 LISTAS COM PROBABILIDADE 0.031250
 14 LISTAS COM PROBABILIDADE 0.007812
 12 LISTAS COM PROBABILIDADE 0.015625
  2 LISTAS COM PROBABILIDADE 0.000488
  5 LISTAS COM PROBABILIDADE 0.003906
```

```
GRAMATICA G

NAO TERMINAIS = STABC
TERMINAIS =AB
FLAGS =FG

CONJUNTO DAS PRODUCOES

1.00000  1 S -----* T+F
0.50000  2 T -----* T+G
0.50000  3 T -----* A+B+A+
1.00000  4 AF-----* A-
1.00000  5 BF-----* B-
1.00000  6 CF-----* B-
1.00000  7 AG-----* A-A+
1.00000  8 BG-----* B-B+C+C+
1.00000  9 CG-----* B-C+

INTERVALO DE APLICACAO DAS PRODUCOES

PRODUCAO  1  0.00000  1.00000
PRODUCAO  2  0.00000  0.50000
PRODUCAO  3  0.50000  1.00000
PRODUCAO  4  0.00000  1.00000
PRODUCAO  5  0.00000  1.00000
PRODUCAO  6  0.00000  1.00000
PRODUCAO  7  0.00000  1.00000
PRODUCAO  8  0.00000  1.00000
```

```
PRODUCAO   9   0.00000   1.00000

GERACAO DAS PALAVRAS

P=0.250000
A A B B B B A A
P=0.500000
A B A
P=0.500000
A B A
P=0.125000
A A A B B B B B B B B B A A A
P=0.250000
A A B B B B A A
P=0.125000
A A A B B B B B B B B B A A A
P=0.250000
A A B B B B A A
P=0.500000
A B A
P=0.500000
A B A
P=0.250000
A A B B B B A A
P=0.062500
A A A A B B B B B B B B B B B B B B B B A A A A
P=0.500000
A B A
P=0.250000
A A B B B B A A
```

```
P=0.031250
A A A A A B B B B B B B B B B B B B B B B B B B B B B B B B A A A A A
P=0.500000
A B A
P=0.500000
A B A
P=0.250000
A A B B B B A A
P=0.250000
A A B B B B A A
P=0.500000
A B A
P=0.500000
A B A
P=0.500000
A B A
P=0.500000
A B A
P=0.500000
A B A
P=0.250000
A A B B B B A A
P=0.031250
A A A A A B B B B B B B B B B B B B B B B B B B B B B B B B A A A A A
P=0.125000
A A A B B B B B B B B B A A A
P=0.500000
A B A
P=0.250000
A A B B B B A A
```

```
P=0.500000
A B A
P=0.062500
A A A A B B B B B B B B B B B B B B B B A A A A
P=0.250000
A A B B B B A A
P=0.500000
A B A
```

E S T A T I S T I C A

AMOSTRA DE 1000 LISTAS

```
235 LISTAS COM PROBABILIDADE 0.250000
529 LISTAS COM PROBABILIDADE 0.500000
133 LISTAS COM PROBABILIDADE 0.125000
 49 LISTAS COM PROBABILIDADE 0.062500
 30 LISTAS COM PROBABILIDADE 0.031250
  8 LISTAS COM PROBABILIDADE 0.007812
  4 LISTAS COM PROBABILIDADE 0.001953
  2 LISTAS COM PROBABILIDADE 0.000488
  6 LISTAS COM PROBABILIDADE 0.003906
  4 LISTAS COM PROBABILIDADE 0.015625
```

```
GRAMATICA G

NAO TERMINAIS = SV*XY
TERMINAIS =DOG,MAN,WOMAN,UGLY,MAD,FAT,NORMALLY,BEAS
TERMINAIS =TLY,JOYOUSLY,LOVES,BITES,EATS,BEATS
FLAGS =FHZG

CONJUNTO DAS PRODUCOES

1.00000  1 S -----* *+FV+G*+F
0.25000  2 *F-----*  -D-O-G- -
0.25000  3 *F-----*  -M-A-N- -
0.25000  4 *F-----*  -W-O-M-A-N- -
0.25000  5 *F-----* X+H*+F
0.25000  6 XH-----*  -U-G-L-Y- -
0.25000  7 XH-----*  -M-A-D- -
0.25000  8 XH-----*  -F-A-T- -
0.25000  9 XH-----* Y+ZX+H
0.33300 10 YZ-----*  -N-O-R-M-A-L-L-Y- -
0.33300 11 YZ-----*  -B-E-A-S-T-L-Y- -
0.33300 12 YZ-----*  -J-O-Y-O-U-S-L-Y- -
0.25000 13 VG-----*  -L-O-V-E-S- -
0.25000 14 VG-----*  -B-I-T-E-S- -
0.25000 15 VG-----*  -E-A-T-S- -
0.25000 16 VG-----*  -B-E-A-T-S- -

INTERVALO DE APLICACAO DAS PRODUCOES
```

```
PRODUCAO  1  0.00000  1.00000
PRODUCAO  2  0.00000  0.25000
PRODUCAO  3  0.25000  0.50000
PRODUCAO  4  0.50000  0.75000
PRODUCAO  5  0.75000  1.00000
PRODUCAO  6  0.00000  0.25000
PRODUCAO  7  0.25000  0.50000
PRODUCAO  8  0.50000  0.75000
PRODUCAO  9  0.75000  1.00000
PRODUCAO 10  0.00000  0.33333
PRODUCAO 11  0.33333  0.66666
PRODUCAO 12  0.66666  1.00000
PRODUCAO 13  0.00000  0.25000
PRODUCAO 14  0.25000  0.50000
PRODUCAO 15  0.50000  0.75000
PRODUCAO 16  0.75000  1.00000

GERACAO DAS PALAVRAS

P=0.01562500
  D O G    B E A T S    W O M A N
P=0.01562500
  D O G    L O V E S    M A N
P=0.01562500
  W O M A N    L O V E S    D O G
P=0.00000042
  F A T    D O G    B I T E S    B E A S T L Y    J O Y O U S L Y    M A D
    D O G
P=0.01562500
  M A N    L O V E S    W O M A N
```

```
P=0.01562500
  W O M A N    L O V E S    M A N
P=0.01562500
  W O M A N    L O V E S    M A N
P=0.01562500
  M A N    B E A T S    D O G
P=0.01562500
  M A N    E A T S    D O G
P=0.00097656
  D O G    B E A T S    F A T    W O M A N
P=0.01562500
  M A N    L O V E S    D O G
P=0.01562500
  D O G    B I T E S    M A N
P=0.00008129
  W O M A N    L O V E S    B E A S T L Y    U G L Y    W O M A N
P=0.00097656
  W O M A N    B I T E S    U G L Y    M A N
P=0.01562500
  M A N    B I T E S    M A N
P=0.00000003
  M A D    M A N    L O V E S    J O Y O U S L Y    J O Y O U S L Y    B E
A S T L Y    M A D    W O M A N
P=0.00097656
  D O G    B I T E S    M A D    D O G
P=0.01562500
  D O G    E A T S    D O G
P=0.00008129
  N O R M A L L Y    M A D    D O G    B E A T S    D O G
P=0.01562500
```

E S T A T I S T I C A

AMOSTRA DE 1000 LISTAS

```
559 LISTAS COM PROBABILIDADE 0.01562500
 11 LISTAS COM PROBABILIDADE 0.00000042
213 LISTAS COM PROBABILIDADE 0.00097656
 59 LISTAS COM PROBABILIDADE 0.00008129
  4 LISTAS COM PROBABILIDADE 0.00000003
 53 LISTAS COM PROBABILIDADE 0.00006103
 28 LISTAS COM PROBABILIDADE 0.00000508
  4 LISTAS COM PROBABILIDADE 0.00000056
 12 LISTAS COM PROBABILIDADE 0.00000381
  9 LISTAS COM PROBABILIDADE 0.00000031
  7 LISTAS COM PROBABILIDADE 0.00000001
 12 LISTAS COM PROBABILIDADE 0.00000676
  7 LISTAS COM PROBABILIDADE 0.00000002
  2 LISTAS COM PROBABILIDADE 0.00000000
  2 LISTAS COM PROBABILIDADE 0.00000004
  3 LISTAS COM PROBABILIDADE 0.00000000
  2 LISTAS COM PROBABILIDADE 0.00000000
  3 LISTAS COM PROBABILIDADE 0.00000023
  3 LISTAS COM PROBABILIDADE 0.00000000
  5 LISTAS COM PROBABILIDADE 0.00000000
  1 LISTAS COM PROBABILIDADE 0.00000001
  1 LISTAS COM PROBABILIDADE 0.00000000
```

```
GRAMATICA G

NAO TERMINAIS = STB
TERMINAIS =01
FLAGS =FG

CONJUNTO DAS PRODUCOES

1.00000  1 S -----* 0-T+F
0.50000  2 T -----* 0-T+G
0.50000  3 T -----* B+
1.00000  4 BF-----* 1-
1.00000  5 BG-----* 0-B+

INTERVALO DE APLICACAO DAS PRODUCOES

PRODUCAO  1  0.00000  1.00000
PRODUCAO  2  0.00000  0.50000
PRODUCAO  3  0.50000  1.00000
PRODUCAO  4  0.00000  1.00000
PRODUCAO  5  0.00000  1.00000

GERACAO DAS PALAVRAS

P=0.25000000
0 0 0 1
P=0.12500000
0 0 0 0 0 1
```

```
P=0.50000000
0 1
P=0.25000000
0 0 0 1
P=0.50000000
0 1
P=0.50000000
0 1
P=0.50000000
0 1
P=0.06250000
0 0 0 0 0 0 0 1
P=0.25000000
0 0 0 1
P=0.50000000
0 1
P=0.50000000
0 1
P=0.50000000
0 1
P=0.25000000
0 0 0 1
P=0.25000000
0 0 0 1
P=0.50000000
0 1
P=0.50000000
0 1
P=0.50000000
0 1
```

```
P=0.50000000
0 1
P=0.12500000
0 0 0 0 0 1
P=0.50000000
0 1
P=0.50000000
0 1
P=0.25000000
0 0 0 1
P=0.50000000
0 1
P=0.50000000
0 1
P=0.25000000
0 0 0 1
P=0.12500000
0 0 0 0 0 1
P=0.12500000
0 0 0 0 0 1
P=0.25000000
0 0 0 1
P=0.25000000
0 0 0 1
P=0.03125000
0 0 0 0 0 0 0 0 0 1
P=0.12500000
0 0 0 0 0 1
P=0.00781250
0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

E S T A T I S T I C A

AMOSTRA DE 1000 LISTAS

```
244 LISTAS COM PROBABILIDADE 0.25000000
105 LISTAS COM PROBABILIDADE 0.12500000
537 LISTAS COM PROBABILIDADE 0.50000000
 52 LISTAS COM PROBABILIDADE 0.06250000
 29 LISTAS COM PROBABILIDADE 0.03125000
 14 LISTAS COM PROBABILIDADE 0.00781250
 12 LISTAS COM PROBABILIDADE 0.01562500
  2 LISTAS COM PROBABILIDADE 0.00048828
  5 LISTAS COM PROBABILIDADE 0.00390625
```

```
GRAMATICA G

NAO TERMINAIS = STB
TERMINAIS =ABC
FLAGS =FG

CONJUNTO DAS PRODUCOES

1.00000  1 S ------* A-T+FB-
0.50000  2 T ------* A-T+G
0.50000  3 T ------* B+
1.00000  4 BF------* C-
1.00000  5 BG------* B-B+

INTERVALO DE APLICACAO DAS PRODUCOES

PRODUCAO  1  0.00000  1.00000
PRODUCAO  2  0.00000  0.50000
PRODUCAO  3  0.50000  1.00000
PRODUCAO  4  0.00000  1.00000
PRODUCAO  5  0.00000  1.00000

GERACAO DAS PALAVRAS

P=0.25000000
A A B C B
P=0.12500000
A A A B B C B
```

```
P=0.50000000
A C B
P=0.25000000
A A B C B
P=0.50000000
A C B
P=0.50000000
A C B
P=0.50000000
A C B
P=0.06250000
A A A A B B B C B
P=0.25000000
A A B C B
P=0.50000000
A C B
P=0.50000000
A C B
P=0.50000000
A C B
P=0.25000000
A A B C B
P=0.25000000
A A B C B
P=0.50000000
A C B
P=0.50000000
A C B
P=0.50000000
A C B
```

```
P=0.50000000
A C B
P=0.12500000
A A A B B C B
P=0.50000000
A C B
P=0.50000000
A C B
P=0.25000000
A A B C B
P=0.50000000
A C B
P=0.50000000
A C B
P=0.25000000
A A B C B
P=0.12500000
A A A B B C B
P=0.12500000
A A A B B C B
P=0.25000000
A A B C B
P=0.25000000
A A B C B
P=0.03125000
A A A A A B B B B C B
P=0.12500000
A A A B B C B
P=0.00781250
A A A A A A A B B B B B B C B
```

```
E S T A T I S T I C A

AMOSTRA DE  1000  LISTAS

  244 LISTAS COM PROBABILIDADE 0.25000000
  105 LISTAS COM PROBABILIDADE 0.12500000
  537 LISTAS COM PROBABILIDADE 0.50000000
   52 LISTAS COM PROBABILIDADE 0.06250000
   29 LISTAS COM PROBABILIDADE 0.03125000
   14 LISTAS COM PROBABILIDADE 0.00781250
   12 LISTAS COM PROBABILIDADE 0.01562500
    2 LISTAS COM PROBABILIDADE 0.00048828
    5 LISTAS COM PROBABILIDADE 0.00390625
```

A P Ê N D I C E   B

```
PAGE    1

// JOB T

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0008        0008        0000

V2 M10   ACTUAL 16K  CONFIG 16K

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(1132PRINTER,PLOTTER)
      DIMENSION Y(30),N(10)
      N(1)=537
      N(2)=244
      N(3)=105
      N(4)=52
      N(5)=29
      N(6)=12
      N(7)=14
      N(8)=5
      N(9)=2
      Y(1)=0.5
      Y(2)=537./1000.
      Y(3)=0.
      Y(4)=0.25
      Y(5)=244./1000.
      Y(6)=0.
      Y(7)=0.125
```

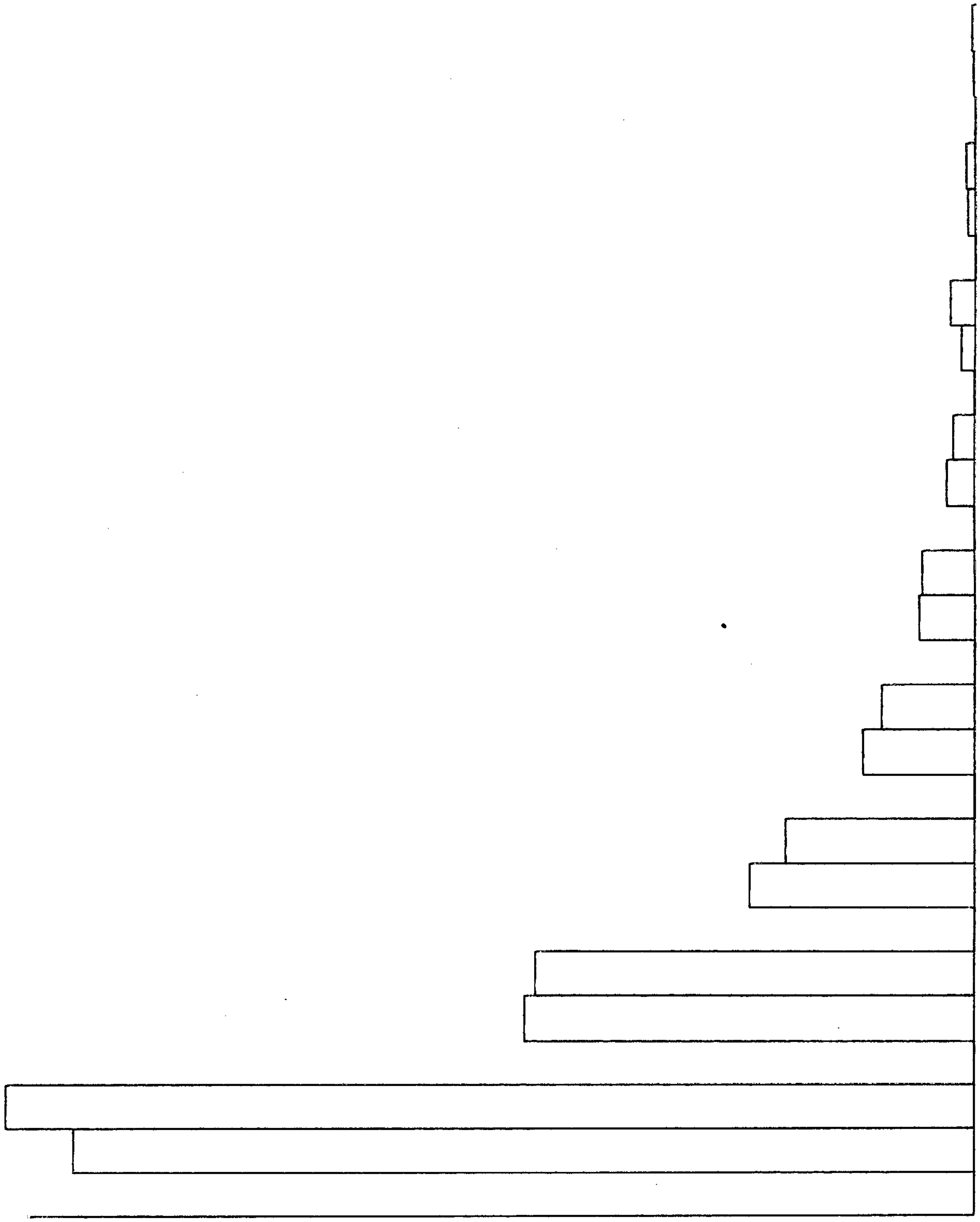
PAGE 2

```
      Y(8)=105./1000.
      Y(9)=0.
      Y(10)=0.0625
      Y(11)=52./1000.
      Y(12)=0.
      Y(13)=0.03125
      Y(14)=29./1000.
      Y(15)=0.
      Y(16)=0.015625
      Y(17)=12./1000.
      Y(18)=0.
      Y(19)=0.007813
      Y(20)=14./1000.
      Y(21)=0.
      Y(22)=0.003906
      Y(23)=5./1000.
      Y(24)=0.
      Y(25)=0.000488
      Y(26)=2./1000.
      WRITE(3,3)
 3    FORMAT(1H1,'NUMERO DE LISTAS  PROBABILIDADE CALCULADA  PROBABILIDA
     *DE REAL'/)
      NABUN=0
      DO 5 I=1,25,3
      NABUN=NABUN+1
 5    WRITE(3,69) N(NABUN),Y(I),Y(I+1)
 69   FORMAT(I10,8X,2F22.7)
      CALL SCALF(0.2816,0.23703,0.,-1.)
```

```
PAGE    3

      CALL FPLOT(1,0.,26.)
      CALL FPLOT(2,0.,0.)
      CALL FPLOT(0,24.,0.)
      CALL FPLOT(1,1.,0.)
      CALL FPLOT(2,1.,0.)
      CALL SCALF(0.2816,11.6,0.,0.)
      DO 50 I=1,26
      X=I-1
      CALL FPLOT(0,X,Y(I))
      CALL FPLOT(0,X+1.,Y(I))
 50   CALL FPLOT(0,X+1.,0.)
      CALL FPLOT(1,0.,-1.)
      STOP
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON       0  VARIABLES      92  PROGRAM    532

END OF COMPILATION

// XEQ
```

NUMERO DE LISTAS	PROBABILIDADE CALCULADA	PROBABILIDADE REAL
537	0.5000001	0.5370000
244	0.2500000	0.2440000
105	0.1250000	0.1050000
52	0.0625000	0.0520000
29	0.0312500	0.0290000
12	0.0156250	0.0120000
14	0.0078130	0.0140000
5	0.0039060	0.0050000
2	0.0004880	0.0020000

```
PAGE    1

// JOB

LOG DRIVE    CART SPEC    CART AVAIL   PHY DRIVE
  0000         0008         0008         0000

V2 M10    ACTUAL 16K   CONFIG 16K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
*IOCS(1132PRINTER,PLOTTER)
      DIMENSION Y(30),N(10)
      N(1)=529
      N(2)=235
      N(3)=133
      N(4)=49
      N(5)=30
      N(6)=4
      N(7)=8
      N(8)=6
      N(9)=4
      N(10)=2
      Y(1)=0.5
      Y(2)=529./1000.
      Y(3)=0.
      Y(4)=0.25
      Y(5)=235./1000.
      Y(6)=0.
```



PAGE 2

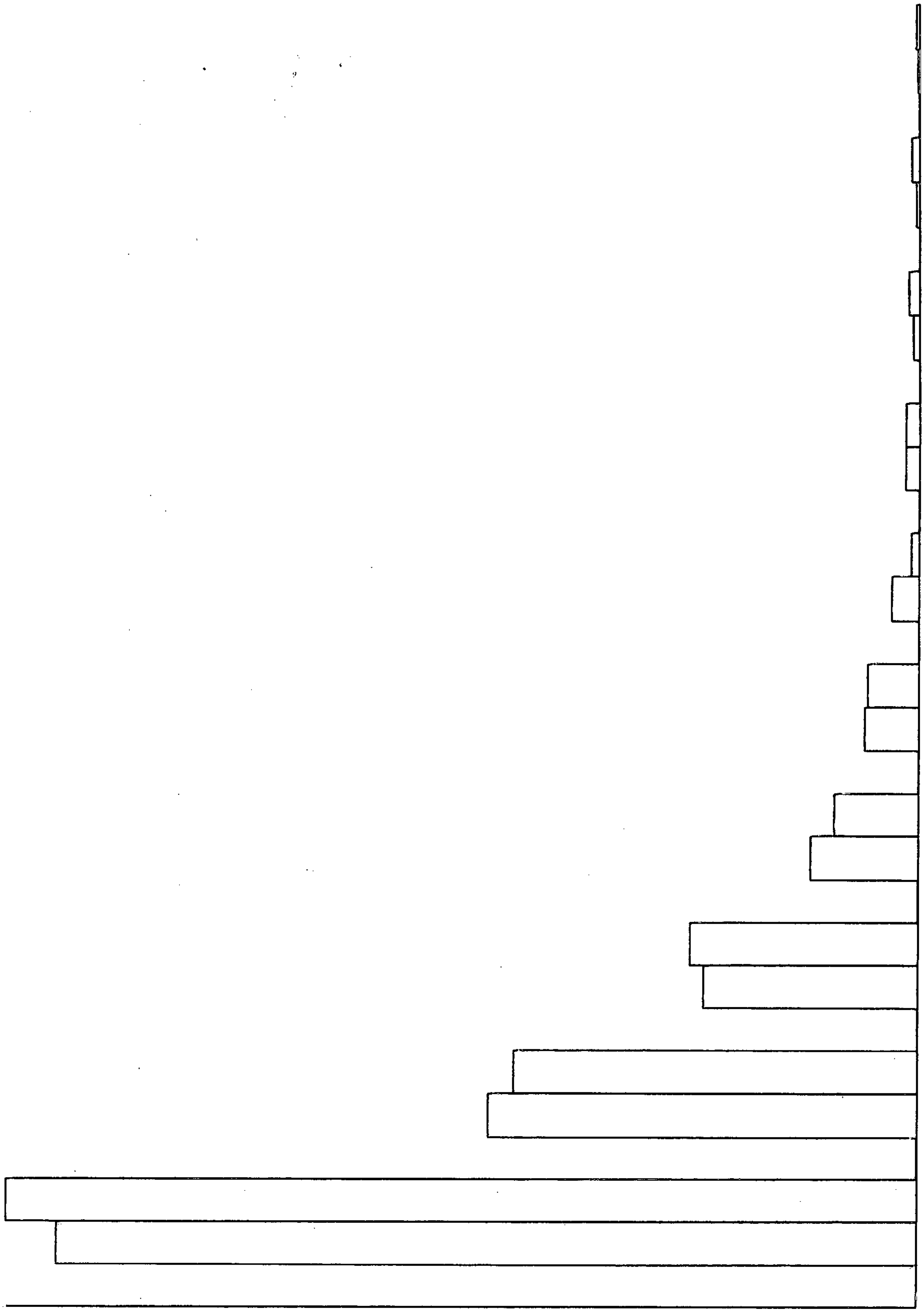
```
      Y(7)=0.125
      Y(8)=133./1000.
      Y(9)=0.
      Y(10)=0.0625
      Y(11)=49./1000.
      Y(12)=0.
      Y(13)=0.03125
      Y(14)=30./1000.
      Y(15)=0.
      Y(16)=0.015625
      Y(17)=4./1000.
      Y(18)=0.
      Y(19)=0.007812
      Y(20)=8./1000.
      Y(21)=0.
      Y(22)=0.003906
      Y(23)=6./1000.
      Y(24)=0.
      Y(25)=0.001953
      Y(26)=4./1000.
      Y(27)=0.
      Y(28)=0.000488
      Y(29)=2./1000.
      WRITE(3,3)
3     FORMAT(1H1,'NUMERO DE LISTAS  PROBABILIDADE CALCULADA  PROBABILIDA
     *DE REAL'/)
      NABUN=0
      DO 5 I=1,28,3
```

```
PAGE   3

      NABUN=NABUN+1
5     WRITE(3,69)N(NABUN),Y(I),Y(I+1)
69    FORMAT(I10,8X,2F22.7)
      CALL SCALF(0.2816,0.23387,0.,-1.)
      CALL FPLOT(1,0.,26.)
      CALL FPLOT(2,0.,0.)
      CALL FPLOT(0,30.,0.)
      CALL FPLOT(1,1.,0.)
      CALL FPLOT(2,1.,0.)
      CALL SCALF(0.2816,11.4552,0.,0.)
      DO 50 I=1,29
      X=I-1
      CALL FPLOT(0,X,Y(I))
      CALL FPLOT(0,X+1,Y(I))
50    CALL FPLOT(0,X+1.,0.)
      CALL FPLOT(1,0.,-1.)
      STOP
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON       0  VARIABLES      92  PROGRAM    560

END OF COMPILATION
```

PAGE 4

// XEQ

NUMERO DE LISTAS	PROBABILIDADE CALCULADA	PROBABILIDADE REAL
529	0.5000001	0.5290000
235	0.2500000	0.2350000
133	0.1250000	0.1330000
49	0.0625000	0.0490000
30	0.0312500	0.0300000
4	0.0156250	0.0040000
8	0.0078120	0.0080000
6	0.0039060	0.0060000
4	0.0019530	0.0040000
2	0.0004880	0.0020000



```
PAGE   1

// JOB T

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0008        0008        0000

V2 M10   ACTUAL 16K  CONFIG 16K

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
*EXTENDED PRECISION
*IOCS(CARD,1132PRINTER)
      DIMENSION X(50),Y(50)
      ITA=0
      READ(2,10) N
 1    READ(2,10) M
      WRITE(3,2)
 2    FORMAT(1H1,'    ')
      ITA=ITA+1
 10   FORMAT(I2)
      READ(2,20) (X(I),I=1,M)
 20   FORMAT(5F10.6)
      READ(2,20) (Y(I),I=1,M)
      T2=0.
      T=0.
      XLY=0.
      TLY=0.
      DO 30 J=1,M
```

```
PAGE   2

       T2=T2+X(J)**2
       T=T+X(J)
       XLY=XLY+ALOG(Y(J))
 30    TLY=TLY+X(J)*ALOG(Y(J))
       DELTA=FLOAT(M)*T2-T**2
       A=(XLY*T2-T*TLY)/DELTA
       B=(FLOAT(M)*TLY-T*XLY)/DELTA
       WRITE(3,40) B,A
 40    FORMAT(/,'  AJUSTAMENTO DE EXPONENCIAL',/,1X,'E**',F15.7,'X+',E15.
      *7,/)
       WRITE(3,101)
 101   FORMAT(/,10X,'QUADRADO DAS DIFERENCAS',/)
       DO 60 I=1,M
       T=(EXP(B*X(I)+A)-Y(I))**2
       WRITE(3,50) T
 50    FORMAT(10X,E15.7)
 60    CONTINUE
       WRITE(3,95)
 95    FORMAT(/,13X,'VALOR REAL',9X,'VALOR AJUSTADO',/)
       DO 90 I=1,M
       T1=EXP(B*X(I)+A)
 90    WRITE(3,96) Y(I),T1
 96    FORMAT(10X,F15.7,2X,F15.7)
       IF(ITA-N)1,770,770
 770   CALL EXIT
       END

FEATURES SUPPORTED
```

```
PAGE   3

 ONE WORD INTEGERS
 EXTENDED PRECISION
 IOCS

CORE REQUIREMENTS FOR
 COMMON       0  VARIABLES     336  PROGRAM     424

END OF COMPILATION

// XEQ
```

```
AJUSTAMENTO DE EXPONENCIAL
E**     -0.7856062X+  0.3081987E 00

        QUADRADO DAS DIFERENCAS

          0.1449367E-01
          0.1075869E-02
          0.1530900E-04
          0.1395790E-04
          0.1991694E-04
          0.1165713E-04
          0.5048172E-05
          0.1873316E-05
          0.4470458E-06
```

VALOR REAL	VALOR AJUSTADO
0.5000000	0.6203896
0.2500000	0.2828004
0.1250000	0.1289126
0.0625000	0.0587639
0.0312500	0.0267871
0.0156250	0.0122107
0.0078130	0.0055661
0.0039060	0.0025373
0.0004880	0.0011566

AJUSTAMENTO DE EXPONENCIAL
E** -0.7309901X+ 0.1387574E 00

QUADRADO DAS DIFERENCAS

0.2818658E-02
0.2649006E-03
0.1019969E-04
0.6137829E-06
0.2364428E-05
0.1743802E-05
0.8581612E-06
0.3487556E-06
0.1273338E-06
0.7864843E-07

VALOR REAL	VALOR AJUSTADO
0.5000000	0.5530910
0.2500000	0.2662757
0.1250000	0.1281936
0.0625000	0.0617165
0.0312500	0.0297123
0.0156250	0.0143044
0.0078130	0.0068866
0.0039060	0.0033154
0.0019530	0.0015961

0.0004880	0.0007684

```
PAGE    1

// JOB T

LOG DRIVE    CART SPEC    CART AVAIL   PHY DRIVE
  0000         0008         0008         0000

V2 M10    ACTUAL 16K   CONFIG 16K

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,PLOTTER)
      DIMENSION VALS(10,2)
   17 READ(2,10) N
   10 FORMAT(I2)
      IF (N) 15,15,16
   15 CALL EXIT
   16 IF (N-10) 12,12,11
   11 N = 10
   12 READ(2,13) ((VALS(I,J),J=1,2),I=1,N)
   13 FORMAT(2F10.7)
      CALL SCALF(8.0/N,10.0,0.0,0.0)
      CALL FGRID(0,0.0,0.0,1.0,N)
      CALL FGRID(1,0.0,0.0,0.1,5)
      DO 14 J=1,2
      CALL FPLOT(1,1.0,VALS(1,J))
      CALL POINT(0)
      DO 14 I=2,N
      CALL FPLOT(2,FLOAT(I),VALS(I,J))
```

```
PAGE   2

  14  CALL POINT(0)
      CALL FPLOT(1,11.0,0.0)
      GO TO 17
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON      0  VARIABLES     46  PROGRAM    204

END OF COMPILATION

// XEQ
```